



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**INFORMAČNÍ PORTÁL PRO STUDENTY**

INFORMATION PORTAL FOR STUDENTS

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. PETR KREJČÍ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. JAROSLAV DYTRYCH, Ph.D.**

**BRNO 2018**

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2017/2018

**Zadání diplomové práce**

Řešitel: **Krejčí Petr, Bc.**

Obor: Informační systémy

Téma: **Informační portál pro studenty**  
**Information Portal for Students**

Kategorie: Informační systémy

**Pokyny:**

1. Seznamte se se současnými komunikačními kanály studentů a existujícími diskusními fóry využívanými studenty FIT VUT.
2. Prostudujte API pro přístup k diskusím na sociální síti Facebook a k dokumentům a kalendářům od firmy Google.
3. Navrhněte nový informační portál pro studenty FIT VUT, který bude zahrnovat diskusi k jednotlivým předmětům a dalším tématům, úložiště dokumentů, studentská skripta a další potřebné funkce. Řešení umožní integraci s diskusemi na sociální síti Facebook a s dalšími službami, které studenti často využívají, a poskytne podporu pro činnost Studentské unie.
4. Implementujte navržené řešení.
5. Vyhodnoťte výsledné řešení a srovnajte s alternativními přístupy

**Literatura:**

- ŠEBEK, Michal. *Implementace diskusní skupiny pro studenty*. Brno, 2007. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. 2007-06-13.
- Další literatura dle doporučení vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a 3.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Dytrych Jaroslav, Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2017

Datum odevzdání: 23. května 2018

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Tato práce se zabývá analýzou současných komunikačních kanálů studentů, návrhem nového webového informačního portálu pro studenty a jeho implementací. Systém integruje sociální síť Facebook, Dokumenty Google a Kalendáře Google. Výsledkem diplomové práce je funkční systém napsaný v jazyce Java, který vznikl na základě analýzy komunikace studentů a požadavků Studentské unie.

## Abstract

This work contains students' communication analysis, the design of new information system for students and its implementation. New system integrates the social network Facebook, the Google Docs and the Google Calendar. The result of this master's thesis is working system written in Java language. The system is based on the students' communication analysis and the requirements of the Students' Union.

## Klíčová slova

informační systém, komunikace, JavaEE, Facebook, Dokumenty Google, Kalendář Google, sazba vzorců, L<sup>A</sup>T<sub>E</sub>X, Spring Security

## Keywords

information system, communication, JavaEE, Facebook, Google Docs, Google Calendar, formula typing, L<sup>A</sup>T<sub>E</sub>X, Spring Security

## Citace

KREJČÍ, Petr. *Informační portál pro studenty*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Dytrych, Ph.D.

# Informační portál pro studenty

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Jaroslava Dytrycha. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Petr Krejčí  
23. května 2018

## Poděkování

Tímto bych chtěl poděkovat Ing. Jaroslavu Dytrychovi za odbornou pomoc na této práci.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Analýza současných komunikačních kanálů studentů</b>	<b>4</b>
2.1	Facebook . . . . .	4
2.2	Fitátor . . . . .	5
2.3	Fituška . . . . .	5
2.3.1	Strukturování dat . . . . .	6
2.3.2	Uživatelské rozhraní . . . . .	7
2.4	Studentská skripta a Dokumenty Google . . . . .	9
<b>3</b>	<b>Požadavky na nový systém</b>	<b>10</b>
3.1	Základní popis . . . . .	10
3.2	Funkční požadavky . . . . .	10
3.3	Nefunkční požadavky . . . . .	13
<b>4</b>	<b>Návrh nového systému</b>	<b>14</b>
4.1	Plán vývoje . . . . .	14
4.2	Role . . . . .	15
4.3	Případy užití . . . . .	17
4.3.1	Vybrané případy užití . . . . .	18
4.4	Použité technologie . . . . .	23
4.5	Architektura . . . . .	27
4.6	Databáze a úložiště souborů . . . . .	28
4.7	Bezpečnost, přihlašování a Spring . . . . .	32
4.8	Plánování úloh . . . . .	33
4.9	Služby s možností zapnutí a vypnutí . . . . .	35
4.10	Služby Google . . . . .	36
4.11	Služba Facebook . . . . .	40
4.12	Služby pro importování uživatelů a předmětů . . . . .	43
4.13	Ostatní služby . . . . .	45
4.14	Uživatelské rozhraní . . . . .	47
<b>5</b>	<b>Implementace klientské části</b>	<b>55</b>

5.1	JSF a Primefaces . . . . .	55
5.2	Editor příspěvků . . . . .	58
<b>6</b>	<b>Implementace serverové části</b>	<b>61</b>
6.1	Služba Facebook . . . . .	61
6.2	Služba Google . . . . .	62
6.3	Migrace předchozí databáze . . . . .	62
<b>7</b>	<b>Testování</b>	<b>66</b>
<b>8</b>	<b>Závěr</b>	<b>69</b>
	<b>Literatura</b>	<b>71</b>
<b>A</b>	<b>Obsah přiloženého CD</b>	<b>73</b>
<b>B</b>	<b>Implementační iterace</b>	<b>74</b>
<b>C</b>	<b>Návrhy uživatelského rozhraní</b>	<b>78</b>

# Kapitola 1

## Úvod

V současné době studenti čím dál více komunikují prostřednictvím Internetu, kde ke komunikaci využívají různé způsoby a různé aplikace. Výsledkem je, že informace, o kterých studenti diskutovali nebo které sdíleli, jsou roztržštěné na různých místech a nebo jsou po nějaké době ztracené.

Pro centralizaci informací a komunikace sice již byl vytvořen systém Fituška<sup>1</sup>, který je dosud dostupný, ale studenti ho přestávají používat a samotný systém přestává fungovat. Fituška postrádá nové funkce a chybí ji administrátoři nebo kdokoliv, kdo by ji udržoval.

Jako náhradu za Fitušku studenti začínají využívat sociální síť Facebook, kde se zakládají diskuzní skupiny pro každý ročník a skupinové konverzace pro řešení projektů či jiných problémů. Pro přípravy na zkoušky se používají sdílené dokumenty společnosti Google (Dokumenty Google). V obou případech se informace neuchovávají a časem se ztratí.

Diplomová práce je rozdělena na 7 hlavních částí. V první části jsou analyzovány typy studentské komunikace a kanály, které studenti používají. Analyzován je především nejčastěji používaný kanál, kterým je Facebook, dosavadní systém Fituška a využití Dokumentů Google. Dále jsou analyzovány kanály, které nejsou tak často využívány ale obsahují funkčnosti, které by mohly podpořit studentskou komunikaci. Druhá část obsahuje sběr požadavků na nový systém, které plynou z analýzy v první části a z požadavků Studentské unie. Třetí část obsahuje návrh nového systému, který ukazuje celkovou strukturu systému (role, architektura a popis služeb) a detailněji specifikuje netriviální požadavky na systém. Čtvrtá část se věnuje implementaci webového uživatelského rozhraní pomocí technologie JavaServer Faces a detailněji popisuje vytvoření editoru příspěvků. Pátá část popisuje implementaci serverové části aplikace a její odchylky od návrhu. Šestá část se věnuje testování nového systému a zpětné vazbě studentů. A poslední část obsahuje závěr se zhodnocením práce.

---

<sup>1</sup>Fituška je studentské diskuzní fórum vytvořené pro studenty Fakulty informačních technologií Vysokého učení technického v Brně. Fituška je dostupná na webové adrese <https://www.fituska.eu/>.

## Kapitola 2

# Analýza současných komunikačních kanálů studentů

Pro dobrý návrh nového systému pro studentskou komunikaci je důležité nejprve zjistit, jakými způsoby studenti komunikují a jaké kanály k tomu využívají.

Co se týče typů komunikace, studenti využívají především diskuzi, sdílení materiálů (studijních textů, vypracovaných projektů, příkladů k procvičování, ...) a vyhledávání v nich. Jako komunikační kanály se využívají zejména současný systém Fituška, sociální síť Facebook, Dokumenty Google a vyhledávač Fitátor. Za zmínku stojí i ne tolik používaná webová stránka Studentská skripta.

### 2.1 Facebook

Nejpoužívanější komunikační kanál je bezpochyby sociální síť Facebook. Studentská unie na této sociální síti pro každý ročník založí 2 skupiny. První skupina je pro diskuzi o tématech ohledně školy. Druhá s názvem teambuilding je určena spíše pro mimoškolní diskuzi. Studenti používají Facebook pro jeho jednoduchost a rozšířenost. Při použití Facebooku je největší šance, že na položený dotaz někdo odpoví.

Skupiny jsou vytvářeny ručně a práva k nim jsou nastavena tak, že obsah vidí pouze členové skupin. Problémem je, že přidávat členy do skupin může jakýkoliv člen skupiny a proto lze říct, že se k obsahu skupin může dostat kdokoli. Postačí poslat žádost o vstup do skupiny a počkat, než nějaký uživatel žádost bez většího zkoumání potvrdí.

Dalším problémem komunikace na Facebooku je jeho přístup k datům. Pro Facebook jsou důležité pouze aktuální příspěvky. Například příspěvky, které uživatel již viděl, se mu na hlavní stránce pravděpodobně nezobrazí. Starší příspěvky musí uživatel vyhledávat buď pomocí vyhledávacího políčka, což v tomto případě znamená vyhledávat v každé skupině zvlášť v nijak strukturovaných datech, nebo uživatel musí posouvat webovou stránku skupiny dolů, dokud se mu hledaný příspěvek nezobrazí. Tento způsob je velmi neefektivní (ale uživateli Facebooku hojně používaný), protože posouvání stránky může být velmi zdoluhavé až nekonečné a navíc tím zvětšujeme obsah stránky, kterou musí náš prohlížeč zobrazit – nároky na výkon při zobrazení stránky rostou.



Posledním velkým problémem jsou podmínky užívání Facebooku. „K obsahu chráněnému právy k duševnímu vlastnictví, jako jsou fotografie a videa (obsah podléhající duševnímu vlastnictví, DV), nám výslovně udělujete následující oprávnění, v souladu s vaším nastavením soukromí a nastavením aplikací: udělujete nám nevýhradní, přenosnou, převoditelnou, celosvětovou bezúplatnou (royalty-free) licenci na použití veškerého obsahu podléhajícího DV, který zveřejníte na Facebooku nebo v návaznosti na něj (Licence k DV). Tato licence k DV končí, jakmile svůj obsah podléhající DV odstraníte ze svého účtu, s výjimkou případů, kdy jste tento obsah sdíleli s ostatními (pokud jej také oni neodstranili)[16].“ Tyto podmínky mohou být problém při sdílení zdrojových kódů, kdy Facebooku udělujeme licenci k neomezenému používání uživatelského díla.

Když to shrneme, tak pozitivní vlastnosti použití Facebooku jsou jeho rozšířenost, už vytvořené skupiny pro diskuzi a jednoduché používání. Negativní vlastnosti jsou nestrukturovaná data, obtížné vyhledávání, licenční podmínky, nemožnost posílat větší soubory a nekontrolovaný přístup k datům (poslední problém by vyřešilo jiné nastavení práv uživatelů ve skupinách).

## 2.2 Fitátor

Fitátor je webová stránka pro vyhledávání ve Facebookových skupinách studentů FIT VUT dostupná na webové adrese [www.fitator.cz](http://www.fitator.cz). Vznik této stránky potvrzuje, že vyhledávání informací ve skupinách za pomoci nástrojů na síti Facebook není použitelné.

Fitátor se snaží data na Facebooku strukturovat pomocí značek<sup>1</sup>. Na začátku každého příspěvku je uvedena jedna nebo více značek, které uživatel musí vložit. Ačkoliv se většinou jako značka použije zkratka předmětu nebo typ zkoušky (půlsemestrální zkouška, semestrální zkouška, projekt, ...), nejsou značky nijak předdefinované. Uživatel musí sám zvolit nejvýstižnější slovo pro značku jeho příspěvku. „Značky jsou následně konsolidovány (změna velkých písmen na malé a odstranění diakritiky)“[15]. Následně při vyhledávání může uživatel vyhledávat nejen podle textu v obsahu příspěvku, ale může se omezit na příspěvky obsahující určité značky. Vyhledávání probíhá ve všech nastavených skupinách Facebooku.

Fitátor takto řeší jeden z problémů použití Facebooku pro studentskou komunikaci ale neřeší ostatní problémy. Naopak problém s přístupem k datům zhoršuje, protože vyhledávat přes něj může úplně kdokoli bez jakéhokoli omezení.

## 2.3 Fituška

Fituška je současný studentský portál pro komunikaci studentů FIT VUT, který je dostupný na adrese <https://www.fituska.eu/>. Tento portál bude nahrazen řešením, které vznikne v rámci této práce. Fituška dle obsahu vznikla někdy v roce 2007. Provozovatelem je podle informací na Fitušce společnost GrowJOB. Toto vidím jako prvním problémem

---

<sup>1</sup>Místo slova „značky“ se v hovorovém jazyce používá anglické slovo „tag“. Značka se uzavírá do hranatých závorek a vkládá před text samotného příspěvku. Například „[koleje]“.

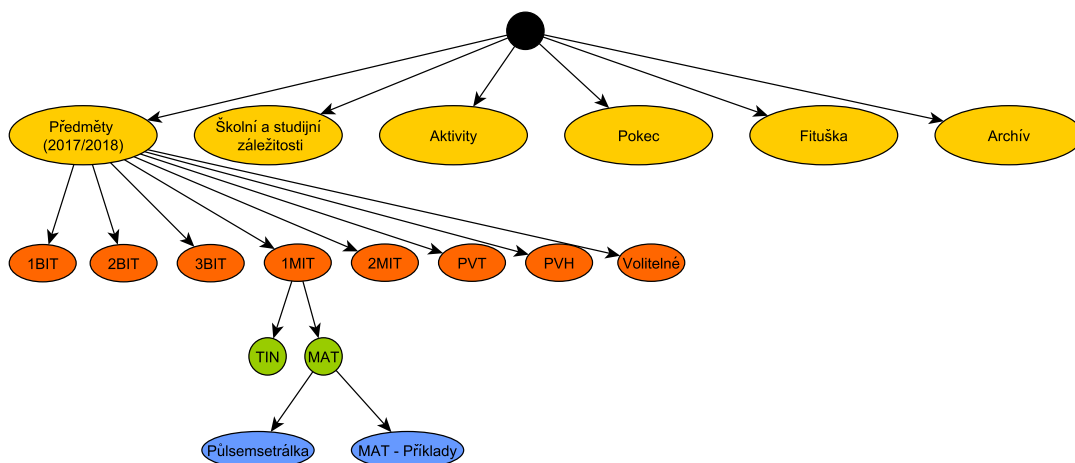
– provozovatelem je společnost, který nemá se školou nic společného. Přičemž v databázi jsou uloženy školní e-maily studentů a obsah týkající se školy.

Bohužel v dnešní době Fituška upadá a pro diskuzi se využívá Facebook. Chybí jí administrátoři a přestává fungovat. Například na začátku akademického roku 2017/2018 Fitušce vypršela platnost certifikátu<sup>2</sup>, web byl pro nevalidní certifikát nedostupný a navíc nebyli do Fitušky vloženi noví studenti, kteří se pak nemohli registrovat a přihlásit.

Kontrola přístupu k datům se realizuje registrací uživatele s platným školním e-mailem. Tímto lze zabezpečit, že přístup k datům budou mít jenom studenti z FIT VUT. Pro ověření, zda je školní e-mail platný musí být e-mail v databázi Fitušky. E-maily se do databáze dostávají manuálním spuštěním skriptu, který získá seznam e-mailů ze školního serveru a zapíše je do databáze. Takto implementované řešení je funkční, ale zvyšuje nároky na údržbu systému.

### 2.3.1 Strukturování dat

Data jsou na portálu strukturována do stromu, kde na nejvyšší úrovni se data dělí do kategorií, dále pak do fór, subfór a vláken. Hlavními kategoriemi jsou *Předměty* pro aktuální rok, *Školní a studijní záležitosti*, *Aktivity*, *Pokec*, *Fituška* a *Archív*.












Obrázek 2.1: Naznačení struktury dat ve Fitušce.

Po každém akademickém roce se celý obsah kategorie *Předměty* přesune do kategorie *Archív* do fóra s názvem akademického roku a vytvoří se nová kategorie *Předměty* s novými prázdnými fóry a subfóry. Tímto vzniká hierarchie akademický rok, předmět a diskuze. Toto řešení bylo asi zvoleno kvůli oddělení zastaralých dat od dat aktuálních. S tímto řešením nesouhlasím a myslím si, že není vhodné, protože obsah předmětů se nemění často a informace z minulých let jsou zpravidla platné i pro aktuální rok. Navíc

<sup>2</sup>V tomto kontextu je míněn certifikát potřebný pro navázání zabezpečeného spojení protokolem HTTPS.

z vlastní zkušenosti, i ze zkušenosti svých spolužáků vím, jak vypadá typické hledání informací o předmětu na Fitušce. Na hlavní stránce si vyhledám předmět a otevřu si jeho stránku v novém okně prohlížeče. Podobně to udělám pro stránky předmětu v archivu za posledních 5 let. Na těchto stránkách si otevřu příslušné vlákno a uvnitř čtu a hledám. Výsledkem je otevřených spousta oken v prohlížeči a spousta zbytečně stráveného času při otevírání stránek předmětu, protože zobrazení seznamu předmětů není moc přehledné.

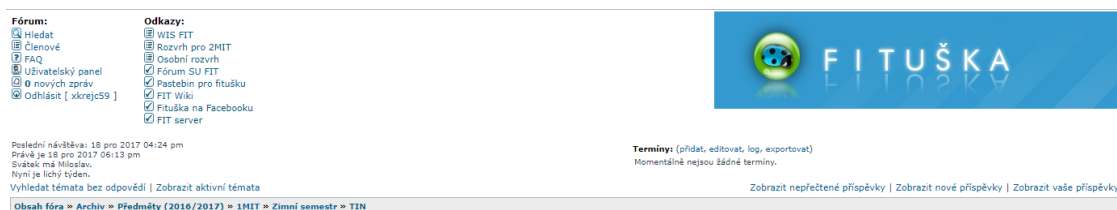
Jak je vidět na obrázku 2.2, většinu předmětů obsahuje položka *Volitelné(V)*, pod kterou je zobrazen jejich seznam. Části seznamu jsou abecedně řazeny ale celek abecedně řazen není. Toto řazení a zobrazení v jednom dlouhém řádku snižuje orientaci a přehlednost.

Obsah fóra » Archiv » Předměty (2015/2016)	
Fórum	
	<b>1BIT</b> Povinné předměty v 1. ročníku bakalářského studia Subfóra: IDA, IEL, IUS, IZP, IMA, INC, IOS, ISU
	<b>2BIT</b> Povinné předměty v 2. ročníku bakalářského studia Subfóra: IAL, IFJ, INM, INP, ISS, IDS, IPK, IPP, IZG, IZU
	<b>3BIT</b> Povinné předměty v 3. ročníku bakalářského studia Subfóra: IIS, IMP, IMS, ISA, ISP, ITU, IBP, ISZ
	<b>1MIT</b> Povinné předměty v 1. ročníku magisterského studia (společné pro všechny obory) Subfóra: MAT, TIN
	<b>2MIT</b> Povinné předměty v 2. ročníku magisterského studia (společné pro všechny obory) Subfóra: SEP, MSZ, DIP
	<b>Povinné volitelné - angličtina (PVA, V) a další jazyky (V)</b> Subfóra: Angličtina (BAN a jiné), BAN1, BAN2, BAN3, BAN4, AEU, FCE, Němčina, Španělština, Ruština
	<b>Povinné volitelné technické (PVT)</b> Subfóra: ICP, IJA, IVH
	<b>Povinné volitelné humanitní (PVH)</b> Subfóra: FIK, HKA, HKO, HPO, HPR, HVR, PRM, RET
	<b>Volitelné (V)</b> Subfóra: IW1, IW2, IW3, IW4, IW5, IBS, IFS, IIZ, IJC, IKR, IMU, INI, IOD, IPA, IRP, ISE, ISJ, ITP, ITS, ITW, ITY, IVS, C2P, C3P, ACH, AGS, AIS, ARC, BIF, BIN, BIO, BIS, BMS, DJA, EIP, EVO, FAV, FLP, FYO, FYO, GAL, GIS, GJA, GMU, GUX, GZN, HSC, JPI, KKO, KRG, KRY, MAR, MEK, MPR, MOG, MUL, NAV, NSB, PBL, PCS, POB, POI, PDS, PES, RGP, PGR, PIS, PKS, PMA, POS, POV, PRL, ROB, ROS, SEN, SFC, SIN, SLO, SNT, SPP, SRI, STI, STM, TAM, THE, VGE, VIN, VIZ, VNV, VYPE, WAP, ZPO, ZRE, ZZN, FSY, IIPD, IKPT, IPSO, MOB, IMAE, IMIE, IPMA / POM, IZEP, IZFI, IZMA, LOG, OPM

Obrázek 2.2: Náhled na zobrazení předmětů na Fitušce.

## 2.3.2 Uživatelské rozhraní

Každá stránka se skládá z horního panelu, hlavního obsahu a spodního panelu. Tyto části jsou graficky odděleny horizontálním šedým pruhem s nápisem „Obsah fóra“. Oddělování obsahu webové stránky pomocí grafických prvků obecně zvyšuje přehlednost a orientaci a zde je toto oddělení správně použito.

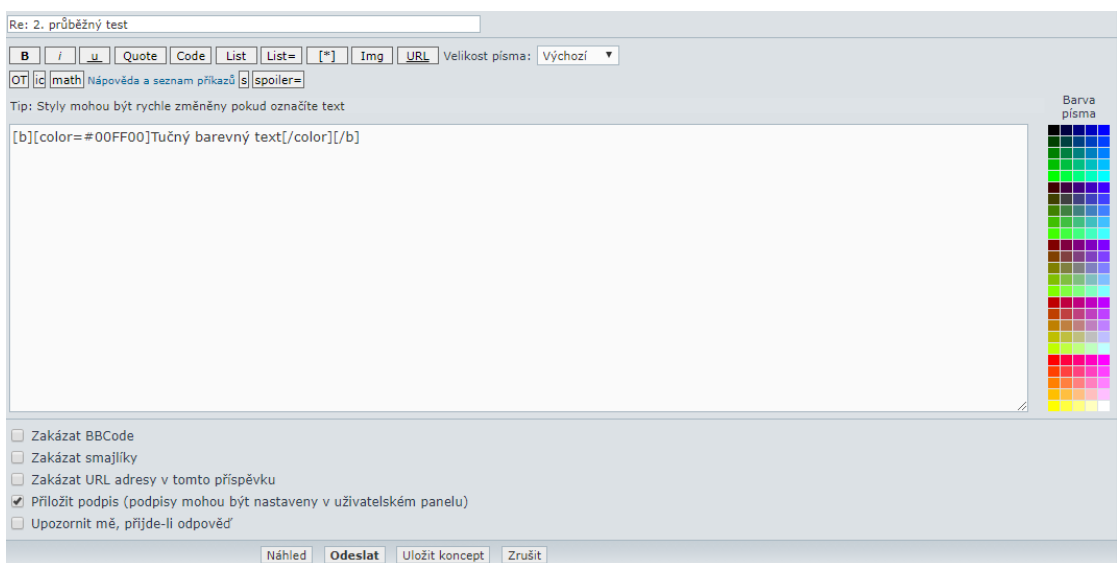


Obrázek 2.3: Horní panel Fitušky.

Horní panel (obrázek 2.3) obsahuje hlavní menu, odkazy mimo fórum, logo Fitušky, informace o termínech a informaci o aktuálním datu. Žádná z těchto částí není nijak graficky oddělena od zbytku. Můj názor je, že ačkoliv je obsah horního panelu zobrazen, není pro uživatele nic víc než obrázek a směsice nepodstatných a nepřehledných znaků. Pravděpodobně proto není ani funkčnost termínů nikým využívána.

Spodní panel obsahuje informace o provozovateli a technologii *phpBB*<sup>3</sup>, na které je fórum postaveno. Na některých stránkách je navíc ve spodním panelu zobrazena informace o uživateli, který je aktuálně přihlášen.

Střední panel, který obsahuje hlavní obsah, je oproti hornímu panelu strukturován správně a přehledně. Zobrazena je vlastně velká tabulka. Grafické prvky oddělují přehledně kategorie i samotné položky v kategoriích. Výtkou, kterou k zobrazení středního panelu mám, je již zmiňovaný seznam volitelných předmětů na hlavní straně a to, že data jsou umístěna po levém a pravém okraji obrazovky a střed obrazovky je většinou prázdný.



Obrázek 2.4: Editor pro přidání příspěvku.

<sup>3</sup>„*phpBB* je oblíbený, bezplatný a open source systém pro vytvoření interaktivního fóra používající serverový skript PHP a podporující širokou škálu databázových systémů.“[20]

Posledním prvkem Fitušky, který chci rozebrat, je editor pro přidávání příspěvků. Tento prvek by měl být na portálu založeném na diskuzi velmi propracovaný a snadno použitelný. Editor nabízí mnoho úprav písma od velikostí a barev, až po formátování pro kód, vložení seznamu a vložení souboru (obrázek 2.4). Na stránce s editorem lze zobrazit i náhled a obsah vlákna, kam příspěvek vkládáme. Editor používá značky HTML pro formátování, které uživatel může vidět v okně s textem nového příspěvku. Toto řešení je výhodné z implementačního hlediska (obsah nového příspěvku se jednoduše uloží a pak zobrazí ve fóru bez velkých úprav) a navíc nemate uživatele (předpokládá se, že studenti HTML znají). Pro tvoření většího příspěvku využívajícího všemožné formátovací úpravy je tento editor velmi vhodný. Co když ale uživatel chce jednoduše odpovědět prostým textem? Potom je tento způsob zbytečně složitý a zdlouhavý. Dle mého názoru by bylo lepší, kdyby měla Fituška 2 editory pro vytvoření příspěvku. Jeden podobný tomuto a druhý zobrazený přímo na stránce vlákna. Druhý editor by byl značně zjednodušený a byl by určen pro jednoduché a krátké odpovědi.

## 2.4 Studentská skripta a Dokumenty Google

Dalším způsobem komunikace, který studenti využívají, je sdílení vlastních studijních textů. Za tímto účelem se používají především 2 komunikační kanály. Prvním jsou Dokumenty Google, druhým je studentský projekt Studentská skripta.

Dokumenty Google jsou zpravidla sdíleny na Facebooku před termínem zkoušky a po zkoušce jsou odkazy na ně ztraceny v záplavě dalších příspěvků a zapomenuty. Dokumentům chybí kontrola přístupu a může je číst a dokonce i upravovat kdokoli, kdo má na tyto dokumenty odkaz. Nastavení práv k dokumentu je možné ale nikdo jej nevyužívá. Domnívám se, že je tomu tak, protože nikde není dostupný seznam všech studentských e-mailových adres, kterým by bylo možné práva přidělit.

Projekt Studentská skripta zabudovává Dokumenty Google do webových stránek <http://fit.fhfstudio.com> a snaží se o kontrolu nad jejich obsahem. Pro každý předmět existuje pouze jeden dokument a ten se postupně edituje (dokument je tedy zachován a není po zkoušce ztracen). Editace funguje ve 2 režimech. První režim je zapnut v období, kdy z předmětu probíhá nějaká zkouška. Uživatelé v prvním režimu mohou provádět libovolné změny v dokumentu. V druhém režimu, který je zapnut mimo období zkoušky, uživatel navrhuje změny formou komentáře v dokumentu, a ty jsou následně schváleny/zamítnuty moderátory Studentských skript [4].

Osobně si myslím, že Studentská skripta nejsou často využívána kvůli jejich malé rozšířenosti mezi studenty a omezenému obsahu. Obsahují pouze 10 dokumentů k předmětům převážně z 3. a 5. semestru bakalářského studia. Navíc jako u samotných dokumentů Google Docs chybí kontrola přístupu.

## Kapitola 3

# Požadavky na nový systém

Výsledkem této práce bude nový informační systém pro studenty zaměřený na podporu komunikace. Nový informační systém nahradí současný systém Fituška a bude spravován Studentskou unií. Nový systém bude pojmenován podle stávajícího systému – Fituška2.

### 3.1 Základní popis

Nový informační systém bude zachovávat všechny používané funkce současného systému a přidá další funkce pro podporu komunikace mezi studenty. Systém bude integrovat sociální síť Facebook – příspěvky budou přenášeny z a na síť Facebook. Systém bude mít kalendář pro vytváření a sdílení událostí. Kvůli rozšířenosti budou využity Kalendáře Google (Kalendář Google má prakticky každý chytrý mobilní telefon a mnoho studentů jej využívá). Pro ukládání rozsáhlejšího a složitějšího obsahu budou využity Dokumenty Google obdobně jako je využívají Studentská skripta. Kontrolu přístupu a zápisu do Dokumentů Google bude spravovat Fituška2. Do příspěvků bude možné vkládat složité vzorce. S nápadem jak tyto vzorce řešit na studentských stránkách přišel M. Šebek, který ve své bakalářské práci [24] použil  $\text{\LaTeX}$ . Proto i tento systém bude používat  $\text{\LaTeX}$  pro sazbu vzorců. Systém bude intuitivní pro používání a přehledný pro snadnou orientaci v něm. Dále systém bude poskytovat podporu pro Studentskou unii ve formě diskuzí, které budou zpřístupněné pouze členům Studentské unie nebo členům vedení Studentské unie. Oproti současnému systému bude použito více automatizace pro snížení nároků na správu. Systém bude provádět automatické rutiny, během kterých bude importovat uživatele a předměty pro nový akademický rok a bude mazat staré příspěvky, události a uživatele. Pro kontrolu přístupu k datům budou využity role, které budou uživatelům přiděleny.

### 3.2 Funkční požadavky

Ze základního popisu nového systému, z analýzy současného systému, z celkové analýzy komunikace studentů a z požadavků Studentské unie byl vytvořen seznam funkčních požadavků. Seznam je rozdělen na *Uživatelské funkční požadavky* (funkce, které budou

využívat přímo uživatelé) a *Systémové funkční požadavky* (funkce, které bude využívat systém nebo které budou systém nastavovat).

## 1. Uživatelské funkční požadavky

- Uživatelské účty
  - přihlásit se
  - odhlásit se
  - vytvoření, editace a smazání účtu
  - základní nastavení (uživatelské jméno, jméno, příjmení, Facebook, Gmail, Skype, heslo, ročník a sebepopis)
  - přidělení a odebrání role
  - nastavení sledovaných značek
  - nastavení sledovaných předmětů
  - nastavení sledovaných předmětů podle WIS<sup>1</sup>
  - nastavení sledovaných školních diskuzí
  - nastavení sledovaných mimoškolních diskuzí
  - nastavení sledovaných kalendářů předmětů
  - nastavení sledovaných ostatních kalendářů
- Fórum
  - zobrazení aktualit, předmětů, školních i mimoškolních diskuzí, písemek, kalendářů, diskuzí Studentské unie, kontaktů, nejbližších událostí, detailu uživatele a zpráv
  - vytvoření, editace a smazání předmětové, školní a mimoškolní diskuze, zkoušky a diskuze Studentské unie
  - přiřazení a odebrání fórum a Facebook značek ke školní, předmětové a mimoškolní diskuzi
  - vyhledávání
  - zaslání hromadné zprávy na Facebook (příspěvek do nastavených skupin)
  - zasílání a přijímání zpráv
  - vytvoření, editace a smazání fórum značek
  - zobrazení kontaktů

---

<sup>1</sup>WIS je oficiální informační systém fakulty, který je dostupný na webové adrese <https://wis.fit.vutbr.cz/FIT/>.

- Diskuze
  - vytvoření, editace, smazání a hodnocení (palců) příspěvků a řešení zkoušky
  - vytvoření, editace a smazání zadání zkoušky
  - zobrazení příspěvků – autor (uživatelské jméno, ikona, role, rozšiřující role a sebe popis), zpráva, doba editace a počet palců
  - zobrazení řešení zkoušky – autor (uživatelské jméno, ikona, role, rozšiřující role a sebe popis), zprávy, doby editace a správnost (palce)
  - sledování a zrušení sledování školní, předmětové a mimoškolní diskuzi a zkoušky
  - vložení anonymního příspěvku
- Příspěvek
  - možný obsah
    - \* text
    - \* obrázek
    - \* soubor
    - \* vzorec
    - \* odkaz na uživatele – mentions
    - \* odkaz na webovou stránku
    - \* anketa
    - \* zdrojový kód
    - \* emotikony
  - formát textu
    - \* barvy, velikost, tučné, podtržené a přeškrtnuté písmo, kurzíva
- Kalendáře
  - vytvoření, smazání a editace kalendáře a událostí
- Facebook
  - přidání, editace a odebrání skupin Facebooku
  - nastavení priority skupin Facebooku
- Ostatní
  - nastavení rutin

## 2. Systémové funkční požadavky

- Rutiny
  - import uživatelů ze školního serveru
  - získání seznamu předmětů ze školního serveru
  - získání odkazů na stránky předmětů ze školního serveru
  - smazání expirovaných uživatelů, událostí, příspěvků a Facebookových skupin



- Facebook
  - prohledávání skupin Facebooku a přenos příspěvků z Facebooku na fórum
  - přenos příspěvků z fóra na Facebook
- Ostatní
  - nastavení práv k používání kalendáře
  - autokorekce českého a slovenského textu
  - import uživatelů na požádání
  - odpojení modulů s cizím API
  - změna certifikátu bez nutnosti restartu serveru
  - vytvoření REST API<sup>2</sup> pro mobilní aplikaci
  - notifikace do prohlížeče

### 3.3 Nefunkční požadavky

- Použití HTTPS a certifikátu.
- Zamezení přístupu uživatelů k obsahu Fitušky 2.0 bez vytvořeného účtu.
- Netransparentní zamezení přístupu uživatelů k obsahu bez patřičného oprávnění.
- Systém zamezí stavu, kdy by nebyl aktivní žádný člen vedení studentské unie. Systém nedovolí smazání uživatele s rolí člen vedení Studentské unie, pokud by jejich počet měl klesnout pod 2. Pokud v systému zůstali méně než 2 aktivní uživatelé s rolí člen vedení Studentské unie, tak se povolí získat tuto roli jakémukoliv studentovi, který je člen Studentské komory AS FIT<sup>3</sup>.

---

<sup>2</sup>REpresential State Transfer je rozhraní pro přístup k datům systému přes protokol HTTP (Hypertext Transfer Protocol). Data jsou často předávána ve formátu XML nebo JSON.

<sup>3</sup>Členy Studentské komory Akademického senátu Fakulty informačních technologií lze získat z webových stránek fakulty <http://www.fit.vutbr.cz/FIT/AS/.cs>

## Kapitola 4

# Návrh nového systému

Po analýze požadavků následoval návrh samotného systému. Návrh systému probíhal postupným tvořením návrhu uživatelského rozhraní a UML diagramů. Vzhledem k tomu, že návrh a vývoj budu provádět pouze já, je použita agilní technika *Just barely good enough*<sup>1</sup>. Při vývoji a návrhu nebyly vytvářeny všechny diagramy a analýzy ale pouze ty, které jsou nutné k pochopení chování a struktury systému. Z tohoto důvodu byl vytvořen pouze diagram databáze, návrh architektury, návrh služeb a návrh uživatelského rozhraní. Dále byla použita technika vytváření tzv. „throw-away“ prototypů<sup>2</sup> pro tvoření jednotlivých částí architektury. Především byly vytvořeny prototypy, ze kterých vznikly všechny služby, a na jejichž základu byly vytvořeny diagramy pro implementaci služeb v systému.

### 4.1 Plán vývoje

Analýza požadavků a návrh jsou rozděleny na 2 hlavní části. V první jsou analyzovány požadavky podle studentské komunikace a podle požadavků Studentské unie. V druhé části bude provedeno uživatelské testování systému a sběr doplňujících požadavků od samotných uživatelů. Samotná implementace bude rozdělena do 4 iterací, kam jsou požadavky rozděleny podle priorit. Poslední (čtvrtá) iterace obsahuje požadavky s nejnižší prioritou, které pravděpodobně nebudou implementovány ale mohou sloužit jako inspirace pro budoucí rozšíření systému. Rozdělení požadavků do implementačních iterací lze nalézt v příloze B.

#### První část

V první části byl navrhnut systém podle analýzy současné komunikace (kapitola 2) a podle požadavků Studentské unie. Výsledkem je návrh uživatelského rozhraní, rozdělení

---

<sup>1</sup> „Model je tak akorát dobrý (just barely good enough), pokud splňuje svůj účel a nic víc, je pochopitelný, jednoduchý, dostatečně přesný, konzistentní a detailní a jeho vytvoření a udržování pozitivně přispívá k projektu.“[2]

<sup>2</sup> „Throw-away prototyp je vytvořen na základě počátečních požadavků ale není použit jako výsledný produkt, i když jeho součásti mohou být znovu použity.“[10]

rolí, rozdělení práv, detailněji specifikované netriviální případy užití, návrh architektury, návrh služeb a databázový model.

Po tomto návrhu bude následovat implementace prvních 3 iterace systému. První iterace je zaměřená na základní funkce diskuzního fóra a správy uživatelů. Po této iteraci bude uživatel mít možnost registrovat se, přihlásit se, vytvořit diskuzi a diskutovat v ní. v druhé iteraci proběhne integrace Kalendářů Google, diskuze bude možné označit značkami a uživatelů budou moci sledovat diskuze s určitou značkou a předměty, které si nastaví. Ve třetí iteraci bude integrována sociální síť Facebook.

## Druhá část

V druhé části proběhne další sběr a analýza požadavků od studentů, kteří si nový systém vyzkoušeli. Pro implementaci budou vybrány požadavky podle užitečnosti a velikosti zásahu do systému. Požadavky, které by měnily základy systému, nebudou přijaty. Přijaté požadavky budou zapracovány do již vytvořeného návrhu, proběhne přepracování webového rozhraní podle nových požadavků a úprava serverové části. Pokud studenti projeví zájem o některé z požadavků ze 4. implementační iterace budou implementovány i ty.

## 4.2 Role

V systému budou mít uživatelé 2 úrovně rolí. První role (základní) je uživateli přiřazena při registraci a zrcadlí jeho roli na fakultě školy. Základní roli musí mít uživatel právě jednu.

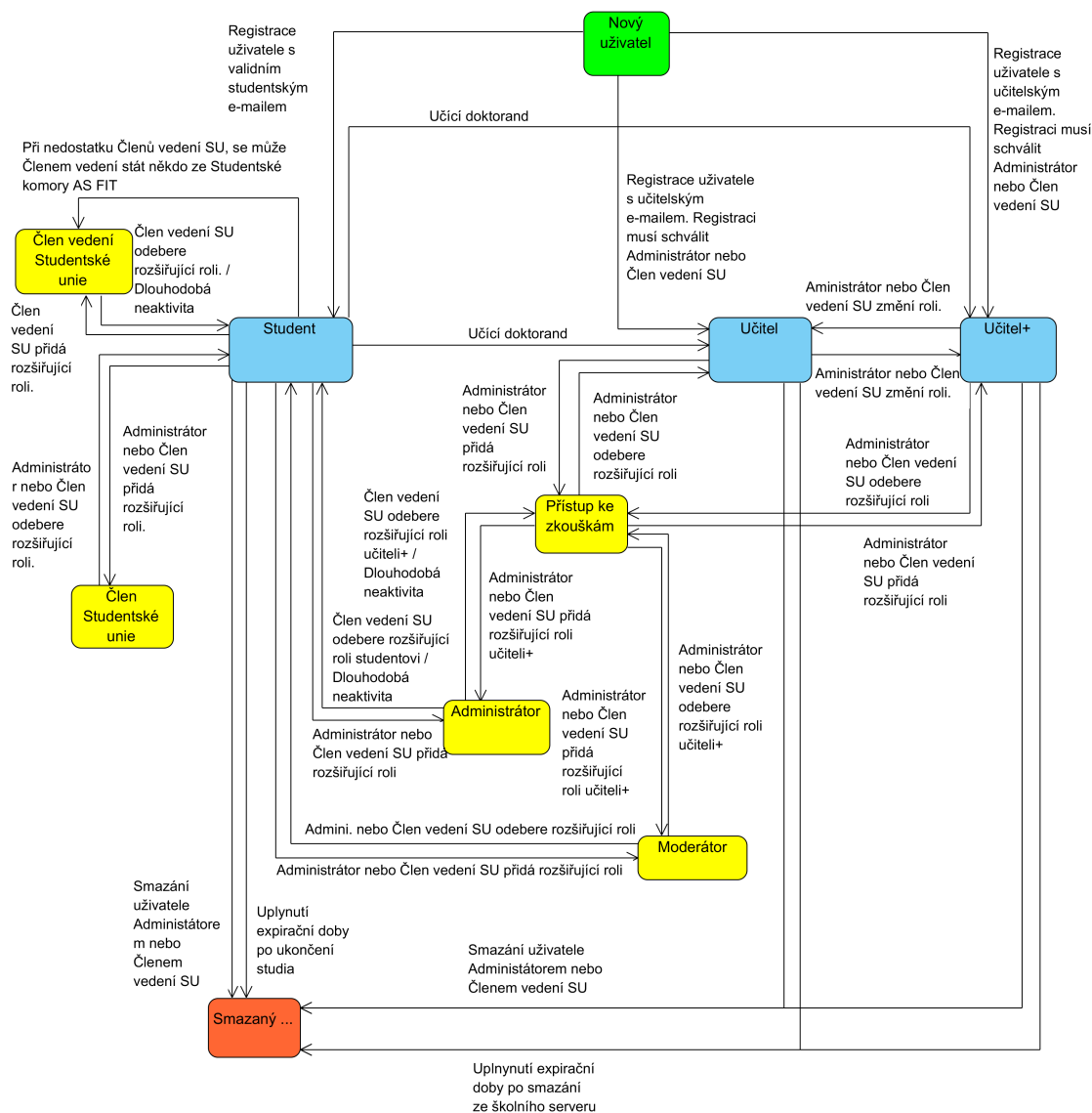
- Student
- Učitel – Vyučující na fakultě se zájmy o školní studentskou činnost
- Učitel+ – Vyučující na fakultě se zájmy o školní i mimoškolní studentskou činnost

Druhá role (rozšiřující) zvyšuje práva uživatele, který se chce zapojit do administrace studentského fóra. Rozšiřujících rolí může mít uživatel více nebo žádnou.

- Přístup ke zkouškám – Tuto roli lze přidělit pouze uživateli se základní rolí Učitel nebo Učitel+. Role zpřístupňuje sekci zkoušek u jednotlivých předmětů.
- Moderátor – Je to role, která může zasahovat do obsahu systému (diskuzí, zkoušek, kalendářů, ...) ale nemůže zasahovat do nastavování systému či rolí uživatelů.
- Řadový člen Studentské unie – Role přidává uživateli přístup k diskuzím a datovému skladu v sekci Studentská unie.
- Administrátor – Druhá nejvyšší role v systému. Její jediné omezení je v přístupu do diskuzí a datového skladu Členů vedení Studentské unie a v tom, že nemůže přiřazovat a odebírat roli Člena vedení Studentské unie.

- Člen vedení Studentské unie – Tato role je nejvyšší v celém systému a obsahuje všechny možná práva. Tato role je nejvyšší z toho důvodu, že Studentská unie bude Fitušku 2.0 spravovat.

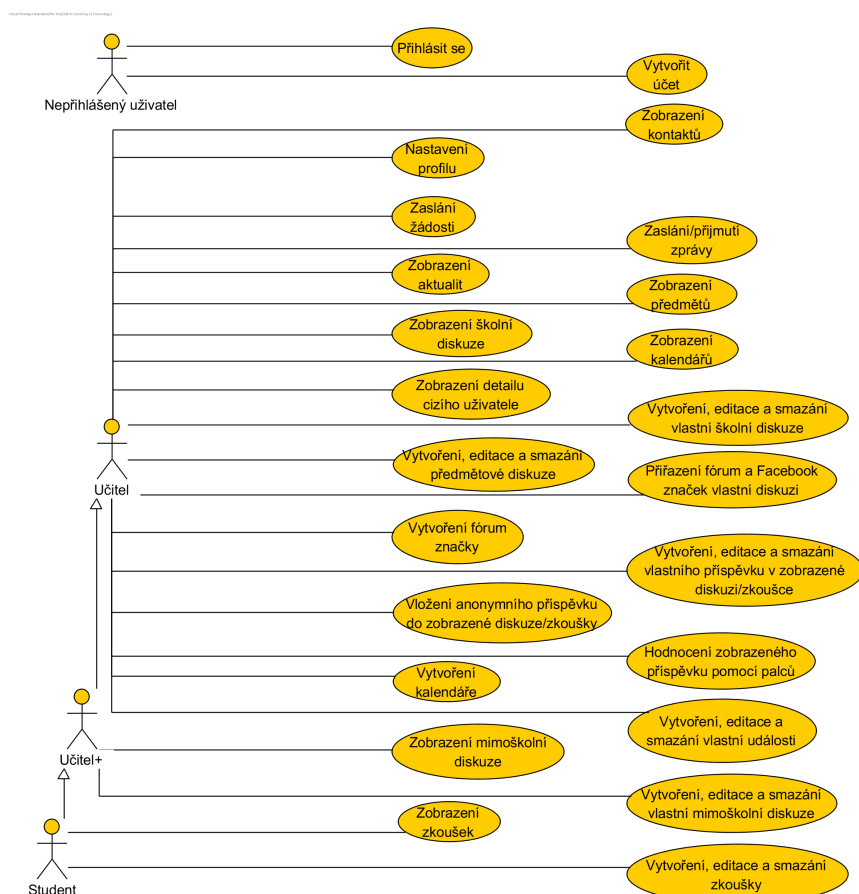
Protože v systému nemůže existovat jakákoliv kombinace rolí, jsou diagramem 4.1 definovány přechody mezi jednotlivými rolemi. Jak je vidět v diagramu, tak například pro to, aby se učitel+ stal administrátorem, musí nejprve získat roli s přístupem ke zkouškám.



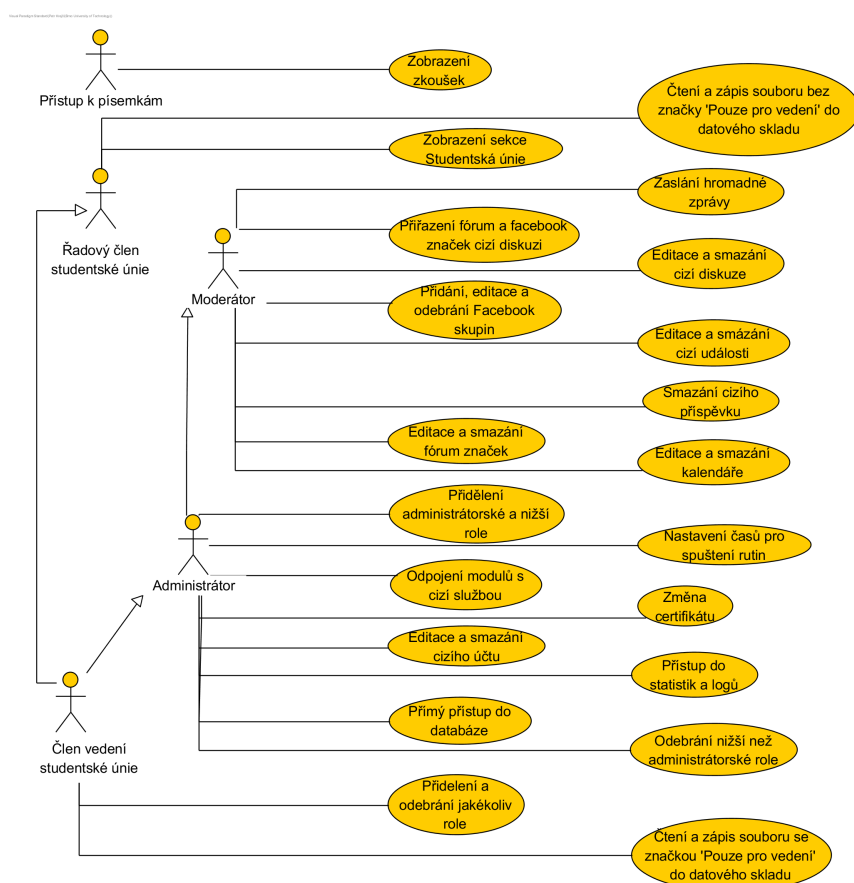
Obrázek 4.1: Diagram přechodů mezi rolemi.

### 4.3 Případy užití

UML diagramy případů užití jsou vytvořeny na základě funkčních požadavků pro uživatele z podkapitoly 3.2. Diagramy slouží především pro přehled rozdělení práv jednotlivým rolím, z toho důvodu jsou vynechány funkční požadavky pro systém. První diagram 4.2 zobrazuje práva základních rolí. Druhý diagram 4.3 zobrazuje práva rozšiřujících rolí. Výsledná práva uživatele vzniknou sloučením práv všech jeho rolí.



Obrázek 4.2: Případy užití základních rolí.



Obrázek 4.3: Případy užití rozšiřujících rolí.

### 4.3.1 Vybrané případy užití

Případy užití jako přihlášení, odhlášení nebo zobrazení aktualit jsou již svým popisem dostatečně vysvětlené a systém se při těchto požadavcích bude chovat, jak by uživatel předpokládal. Naopak případy jako Registrace studenta nebo Vytvoření diskuze skrývají spoustu kroků, a proto jsou v této kapitole detailněji vysvětleny.

<b>Případ užití:</b>	Registrace studenta
<b>Účastníci:</b>	Neregistrovaný student a systém
<b>Vstupní podmínky:</b>	Uživatel je podle dat ze školního serveru identifikován jako student.
<b>Kroky:</b>	<ol style="list-style-type: none"> <li>1. Student vyplní registrační formulář.</li> <li>2. Systém ověří zadané údaje v databázi.</li> <li>3. Systém vytvoří nový uživatelský účet pro studenta, přidělí mu roli student a na zadaný e-mail zašle aktivační odkaz pro potvrzení vlastnictví e-mailu.</li> <li>4. Student pomocí obdržného odkazu aktivuje účet.</li> </ol>
<b>Výjimky:</b>	<p>[2] Systém nenalezne uživatele. Registrace proběhla neúspěšně.</p> <p>[2] Uživatel se zadaným jménem již existuje. Registrace proběhla neúspěšně.</p> <p>[2] Heslo a ověření hesla se neshodují nebo je heslo moc krátké. Registrace proběhla neúspěšně.</p> <p>[2] Uživatel se zadaným e-mailem je již registrován. Registrace proběhla neúspěšně.</p>
<b>Výstupní stav:</b>	Vytvořený uživatelský účet.

<b>Případ užití:</b>	Registrace učitele
<b>Účastníci:</b>	Neregistrovaný učitel, systém a členové vedení Studentské unie
<b>Vstupní podmínky:</b>	Uživatel je podle dat ze školního serveru identifikován jako učitel.
<b>Kroky:</b>	<ol style="list-style-type: none"> <li>1. Učitel vyplní registrační formulář.</li> <li>2. Systém ověří zadané údaje v databázi.</li> <li>3. Systém vytvoří hlasování o schválení vytvoření účtu učitele.</li> <li>4. Alespoň 3 členové vedení Studentské unie schválí vytvoření účtu.</li> <li>5. Systém vytvoří nový účet pro učitele, přiřadí mu roli učitele a na zadaný e-mail zašle aktivační odkaz pro potvrzení vlastnictví e-mailu.</li> <li>6. Učitel pomocí obdrženého odkazu aktivuje účet.</li> </ol>
<b>Alternativní tok:</b>	<p>[3.1] Vyprší lhůta pro hlasování a hlasování obsahuje alespoň jeden pozitivní hlas. Systém uzavře hlasování jako pozitivní a přejde k bodu [5].</p> <p>[3.1] Funkce schvalování učitelské registrace je vypnuta. Systém přejde k bodu [5]</p>
<b>Výjimky:</b>	<p>[2] Uživatel se zadaným jménem již existuje. Registrace proběhla neúspěšně.</p> <p>[2] Heslo a ověření hesla se neshodují nebo je heslo moc krátké. Registrace proběhla neúspěšně.</p> <p>[2] Uživatel se zadaným e-mailem je již registrován. Registrace proběhla neúspěšně.</p> <p>[4] Alespoň jeden člen vedení Studentské unie hlasoval proti. Vytvoření účtu je zamítnuto.</p>
<b>Výstupní stav:</b>	Vytvořený uživatel.



<b>Případ užití:</b>	Vytvoření diskuze
<b>Účastníci:</b>	Registrovaný uživatel a systém
<b>Vstupní podmínky:</b>	Uživatel má oprávnění pro vytvoření dané diskuze.
<b>Kroky:</b>	<ol style="list-style-type: none"> <li>1. Uživatel vyplní název diskuze.</li> <li>2. Uživatel přiřadí fórum značky.</li> <li>3. Uživatel přiřadí Facebook značky a skupiny, kde se mají vyhledávat.</li> <li>4. Systém vytvoří novou diskuzi.</li> </ol>
<b>Alternativní tok:</b>	
<b>Výjimky:</b>	<p>[3] Vybraná asociace Facebook značka a skupina již existuje. Značka nebude přidána.</p> <p>[4] Diskuze se zadaným jménem už v kategorii (školní, předmětová a mimoškolní) již existuje.</p>
<b>Výstupní stav:</b>	Vytvořená nová diskuze.

<b>Případ užití:</b>	Vytvoření příspěvku
<b>Účastníci:</b>	Registrovaný uživatel, systém
<b>Vstupní podmínky:</b>	Uživatel má právo vložit příspěvek
<b>Kroky:</b>	<ol style="list-style-type: none"> <li>1. Uživatel vyplní text příspěvku.</li> <li>2. Uživatel text naformátuje pomocí zobrazených tlačítek nebo značek jazyka HTML.</li> <li>3. Uživatel může vložit obrázek nebo jiný soubor.</li> <li>4. Uživatel může vložit odkazy na jiné uživatele.</li> <li>5. Uživatel může vložit vzorce pomocí zobrazených tlačítek nebo může vložit přímo zdrojový kód v jazyce <math>\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}</math>.</li> <li>6. Uživatel může vložit odkaz na webovou stránku.</li> <li>7. Uživatel zvolí, zda má být příspěvek anonymní.</li> <li>8. Systém uloží příspěvek a požadavek na nahrání příspěvku na Facebook do fronty požadavků.</li> </ol>
<b>Alternativní tok:</b>	
<b>Výjimky:</b>	<p>[3] Obrázek nebo soubor se nepodaří nahrát na fórum.</p> <p>[8] Příspěvek přesahuje maximální délku nebo obsahuje nevalidní značky jazyka HTML.</p>
<b>Výstupní stav:</b>	Příspěvek je vložen do fóra.

<b>Případ užití:</b>	Přenos příspěvku z fóra na Facebook pod účtem Fitušky 2.0
<b>Účastníci:</b>	Systém a Facebook
<b>Vstupní podmínky:</b>	Ve frontě požadavků je příspěvek. Služba Facebook je zapnutá.
<b>Kroky:</b>	<ol style="list-style-type: none"> <li>1. Systém vybere příspěvek z fronty požadavků.</li> <li>2. Systém zjistí, které skupiny a které značky jsou asociované s diskuzí, kam byl vložen příspěvek.</li> <li>3. Systém nahraje příspěvek do skupin na Facebooku pod svým účtem. Před příspěvek do hranatých závorek přidá všechny značky, které našel.</li> <li>4. Facebook přidá příspěvek do skupin a vrátí odkazy vložených příspěvků.</li> </ol>
<b>Alternativní tok:</b>	
<b>Výjimky:</b>	[2] Žádné asociace k příspěvku nejsou nastaveny.
<b>Výstupní stav:</b>	Příspěvek je vložen do asociovaných skupin na Facebooku.

<b>Případ užití:</b>	Stáhnutí příspěvku z Facebooku na fórum
<b>Účastníci:</b>	Systém a Facebook
<b>Vstupní podmínky:</b>	V systému jsou přidány Facebookové skupiny, které má systém prohledávat.
<b>Kroky:</b>	<ol style="list-style-type: none"> <li>1. Uplynula nastavená doba od posledního prohledávání.</li> <li>2. Systém zjistí, kdy byly skupiny naposled prohledávány.</li> <li>3. Systém vyhledá ve skupinách všechny příspěvky s datem vytvoření novějším, než je zjištěné datum posledního prohledávání.</li> <li>4. Příspěvky, které na svém začátku obsahují značky, a které jsou asociované s nějakou fórem diskuzí, systém přidá do databáze a do diskuze. Jako autor je uveden uživatel asociovaný s účtem Facebooku, který je uveden jako autor příspěvku na Facebooku.</li> </ol>
<b>Alternativní tok:</b>	4.1 Facebook účet autora příspěvku na Facebooku není uveden u žádného uživatele na fóru. Příspěvek je přidán bez autora.
<b>Výjimky:</b>	
<b>Výstupní stav:</b>	Příspěvky s validními značkami jsou staženy na fórum.

<b>Případ užití:</b>	Aktualizace příspěvků ze sítě Facebook
<b>Účastníci:</b>	Systém a Facebook
<b>Vstupní podmínky:</b>	V systému jsou přidány Facebookové skupiny, které má systém prohledávat. Uplynula nastavená doba od posledního prohledávání.
<b>Kroky:</b>	1. Systém zjistí, kdy byly skupiny naposled prohledávány. 2. Systém vyhledá ve skupinách všechny příspěvky s datem editace novějším než je zjištěné datum posledního prohledávání. 3. Obsah příspěvku se stejným Facebook ID je nahrazen nalezeným příspěvkem.
<b>Alternativní tok:</b>	3.1 Datum poslední úpravy příspěvku na fóru je novější než datum úpravy příspěvku na Facebooku. Je zachován příspěvek na fóru.
<b>Výjimky:</b>	
<b>Výstupní stav:</b>	Příspěvky jsou aktualizovány.

<b>Případ užití:</b>	Editace příspěvku na fóru.
<b>Účastníci:</b>	Uživatel, systém a Facebook
<b>Vstupní podmínky:</b>	Příspěvek je nahrán do některých skupin na Facebooku.
<b>Kroky:</b>	1. Uživatel upraví příspěvek obdobně jako při jeho vytváření. 2. Systém vyhledá skupiny, ve kterých je příspěvek nahrán. 3. Systém upraví příspěvky ve všech skupinách pomocí svého profilu.
<b>Alternativní tok:</b>	3.1 Systém není autorem příspěvku na Facebooku. Systém uchová změněný příspěvek na fóru a příspěvky na Facebooku budou nezměněny.
<b>Výjimky:</b>	
<b>Výstupní stav:</b>	Příspěvky jsou aktualizovány.

## 4.4 Použité technologie

Jako první technologie musí být vybrán samotný programovací jazyk, ve kterém bude webová aplikace naprogramována. V této oblasti se nabízí spousta jazyků a většina z nich poskytuje přímo framework<sup>3</sup> pro jednodušší práci při vytváření webových aplikací. Nejvhodnějšími kandidáty pro tuto aplikaci jsou jazyk PHP (lze použít některý jeho framework) a jazyk Java s technologiemi Java Enterprise Edition (Java EE). Obě možnosti poskytují tzv. podnikové technologie, které lze při vývoji webové aplikace použít, například objektově-relační mapování nebo vkládání závislostí (anglicky Dependency

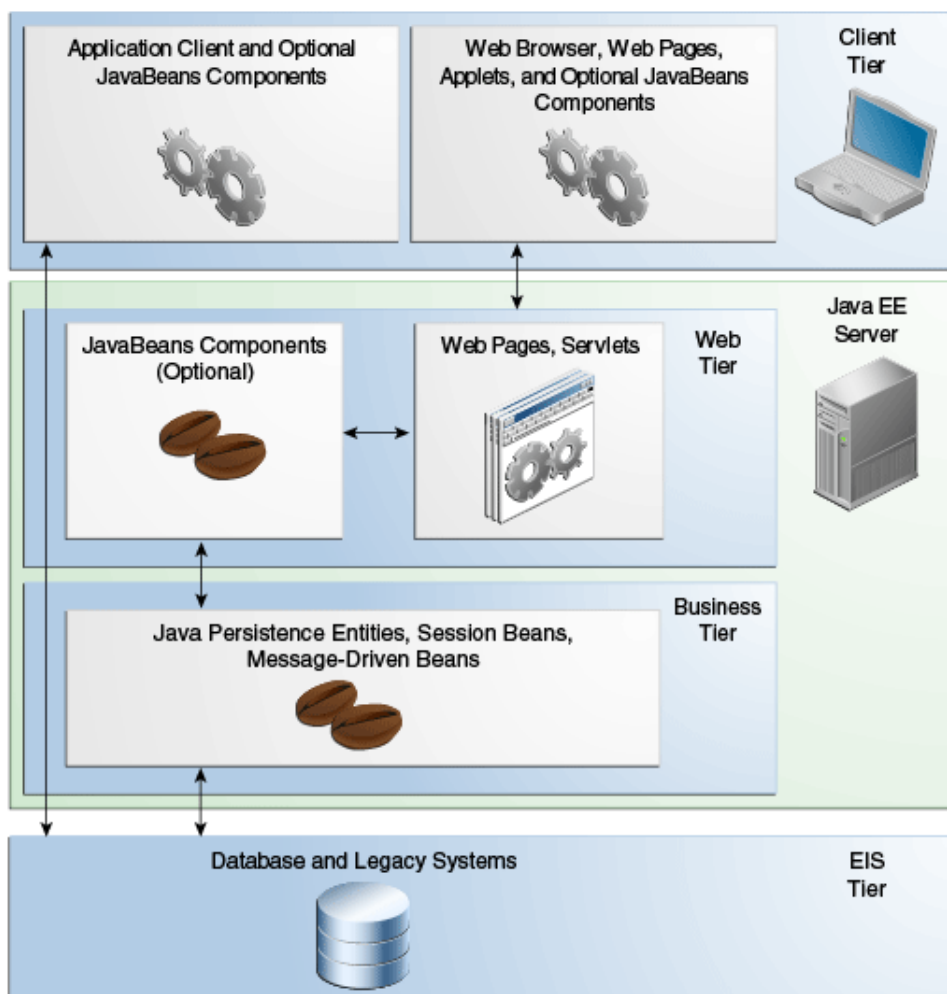
<sup>3</sup> „Framework (aplikační rámec) je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů.“ [23] Odborná veřejnost obecně používá anglický název „framework“, proto i já budu dále v této práci používat termín „framework“ namísto českého překladu „aplikační rámec“, který není tak známý.

injection). Dle různých experimentů uvedených ve článku [9] je v některých oblastech lepší PHP a v některých Java, ale při celkovém zhodnocení jsou jazyky výkonnostně srovnatelné.

Je běžnou praxí pro menší webový projekt zvolit první možnost. PHP je dynamický jazyk a tak není nutné aplikaci neustále sestavovat, pokud ji chce programátor otestovat. PHP může být navíc spouštěno na webovém serveru Apache, který je součástí takřka každého stroje, který poskytuje webové služby. Výhodami PHP jsou tedy rychlý vývoj a snadné nasazení.

Pro větší projekty se naopak vybírá druhá možnost. Java se svými pokročilými technologiemi je vhodná pro systémy, které se skládají ze spousty částí a které komunikují s jinými systémy. Navíc technologie Java EE oddělují uživatelské rozhraní od serverové části a tím je struktura aplikace přehlednější.

Pro tento projekt jsem se rozhodl pro jazyk Java a to proto, že chci využít jasnou architekturu (obrázek 4.4) webové aplikace Java EE, framework Primefaces, který poskytuje spoustu hotových komponent pro tvoření uživatelského rozhraní, a technologii Spring Security pro zajištění bezpečnosti aplikace.



Obrázek 4.4: Architektura webové aplikace v Java EE s technologií JavaServerPages.[8]

Aplikacím v Java EE se často vytýká jejich složité nasazení a složitá konfigurace serveru, na kterém běží, zejména oproti aplikacím ve zmiňovaném PHP. Jsem toho názoru, že vhodným výběrem technologií lze tomuto předejít a vytvořit aplikaci s jednoduchou možností nasazení a minimální konfigurací.

Pro aplikaci v Java EE je nutné nejprve vybrat server, na kterém poběží. Servery pro aplikace v Java EE lze rozdělit na 2 kategorie. První kategorie obsahuje tzv. aplikační servery. „Aplikační server implementuje rozhraní platformy Java EE a poskytuje standardní služby Java EE. Aplikační servery poskytují data klientům, stejně jako webový server poskytuje webové stránky prohlížeči. Webový kontejner je rozhraní mezi webovými komponentami a webovým serverem. Kontejner spravuje životní cykly komponent, zpracovává požadavky na komponenty a poskytuje informace o požadavcích.“ [5] Pro spuštění aplikace v Java EE je tedy možné využít aplikační server, který poskytuje všechny standardní technologie a služby nebo webový kontejner, který poskytuje pouze

základní technologie (jako Servlet, JavaServerFaces, ...) pro spuštění webové aplikace. Díky omezenosti technologií a možností webového kontejneru je samozřejmě jednodušší jeho spuštění a konfigurace oproti aplikačnímu serveru. Z tohoto důvodu jsem si vybral webový kontejner Tomcat 9.0, který poskytuje většinu technologií, které chci využít.

Z komponent architektury aplikace v Java EE (obrázek 4.4) Tomcat poskytne webovou vrstvu (na obrázku označenou jako „WebTier“). Zbývá vybrat technologie pro aplikační vrstvu (na obrázku „Business tier“) a datovou vrstvu („EIS Tier“). Pro tvorbu aplikační vrstvy je doporučená technologie Enterprise Java Beans (zkráceně EJB)<sup>4</sup>. Jako další možnost lze použít poměrně novou technologii Context Dependency Injection (zkráceně CDI)<sup>5</sup>. Vzhledem k tomu, že technologie, kterou používá kontejner Tomcat pro správu svých komponent zastarala a je doporučeno nahradit ji technologií CDI, jsem vybral CDI i pro vytvoření aplikační vrstvy. Výhody, které poskytuje EJB (jako komunikace mezi různými systémy) v aplikaci nebudou využity, což je další důvod pro ponechat pouze CDI.

Poslední chybějící částí je databáze a technologie pro objektově-relační mapování. Jako databázi jsem vybral MySQL. Důvody k výběru právě této databáze jsou její velká popularita, rozšířenost, multiplatformnost, což usnadňuje budoucí nasazení aplikace, a její bezplatná licence GPL. Pro komunikaci mezi databází a aplikací a pro objektově-relační mapování jsem vybral technologii Hibernate. Jedná se také o velmi rozšířenou a navíc dobře zdokumentovanou technologii.

Tímto byly vybrány všechny technologie nutné pro spuštění a nasazení aplikace. Dále byl vybrán framework Primefaces, framework Spring Security a nástroj Maven. Primefaces poskytuje spousty hotových komponent pro tvoření webového uživatelského rozhraní technologie JavaServer Faces (JavaServer Faces je obsažena v kontejneru Tomcat). „Spring Security poskytuje služby pro zabezpečení aplikací v Java EE a je především zaměřen na autentizaci a autorizaci.“[1]. Nástroj Maven<sup>6</sup> je použit pro sestavení aplikace a správu závislostí systému.

V této kapitole jsou uvedeny pouze hlavní technologie, které se týkají celého projektu. Použití dalších technologií a knihoven je uvedeno v kapitolách týkajících se návrhu dílčích částí systému.

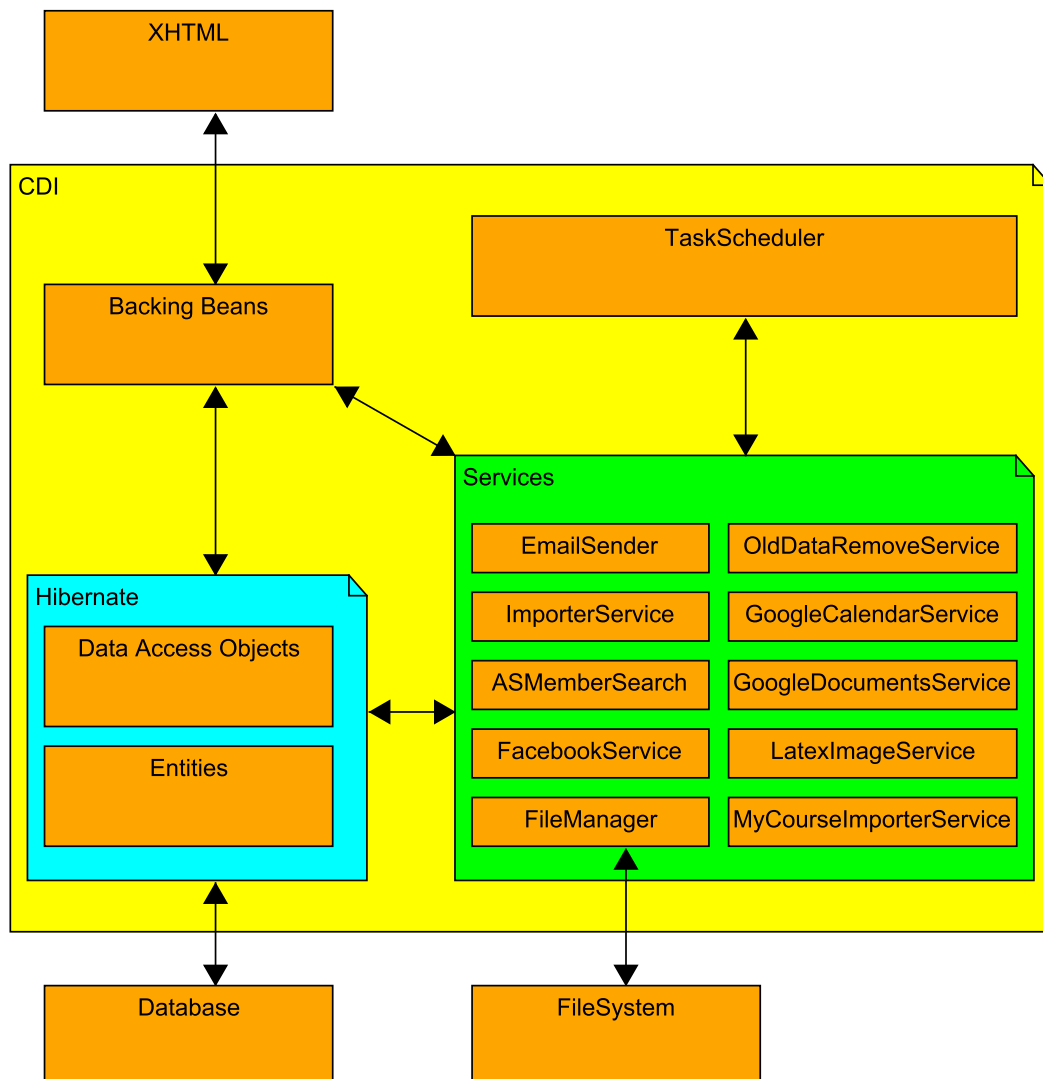
---

<sup>4</sup> „Enterprise Java Bean je technologie pro tvoření komponent pro aplikační vrstvu. Technologie spravuje životní cyklus komponent, jejich závislosti a dovoluje použití komponent i jiným systémům (mimo aplikaci nebo i mimo server). EJB dovoluje vytvořit 3 typy komponent – Session Bean, Entity Bean a Message-driven beans.“[3]

<sup>5</sup> Context Dependency Injection je technologie pro správu životních cyklů komponent a pro vkládání závislostí. Pomocí CDI lze ovlivnit životní cyklus komponenty nastavením jejího rozsahu.

<sup>6</sup> „Maven je nástroj společnosti Apache Software Foundation určený k řízení a automatizaci sestavování aplikací. Ačkoliv je možné použít tento nástroj pro projekty psané v různých programovacích jazycích, podporován je převážně jazyk Java. Pomocí konfiguračního souboru (často pojmenovaný pom.xml) lze popsat způsob sestavení softwaru a jeho závislosti. Nástroj na základě tohoto souboru stáhne závislosti z úložiště a aplikace sestaví.“[18]

## 4.5 Architektura



Obrázek 4.5: Navržená architektura

Při návrhu architektury aplikace jsem se řídil architekturou aplikace v Java EE od Oracle (obrázek 4.4) a vybranými technologiemi, které definují strukturu aplikace pro jejich použití.

Jak bylo dříve řečeno, tak pro klientskou část je použita technologie JavaServer Faces (dále JSF) a framework Primefaces. JSF rozděluje klientskou část na webové stránky v jazyce XHTML<sup>7</sup> a tzv. Backing Beans<sup>8</sup>.

Další technologií, která definuje část architektury, je Hibernate, který používá návrhový vzor Data Access Object<sup>9</sup>, který v kontextu této aplikace definuje vytvoření entit (třídy, které se plní daty z databáze) a tříd, které poskytují přístup k těmto entitám a vytváří dotazy na databázi (přístupové třídy). Pro každou entitu bude vytvořena odpovídající přístupová třída obsahující metody, které vrací entitu, seznam entit nebo informaci o entitě. Tento návrh řeší rozhodování, která metoda bude patřit které přístupové třídě.

Důležitou součástí systému jsou služby, které jsou součástí aplikační vrstvy. Tyto služby často komunikují nebo zpracovávají data mimo aplikaci nebo jsou to moduly s funkcemi, které se vykonávají periodicky tzv. rutiny. Součástí služeb je i FileManager, což je služba, která se stará o ukládání souborů na pevný disk serveru. Pro plánování a spouštění rutin je určena služba, která dokáže vykonat jednu danou funkci v nastavený čas nebo ji vykonávat opakovaně s danou frekvencí.

Poslední nezmíněnou součástí architektury jsou perzistentní úložiště dat. Je to databáze MySQL, jak bylo zmíněno dříve, a je to souborový systém serveru, kam se budou ukládat obrázky a obecně soubory, které uživatelé uloží do systému.

## 4.6 Databáze a úložiště souborů

Protože uživatel může do systému vkládat obrázky a jiné soubory, je nutné je ukládat. Jako řešení tohoto problému se nabízí 3 možnosti. První možností je ukládat soubory do databáze, druhou možností je ukládat soubory do souborového systému serveru a poslední možností je využít externí úložiště, například FileCloud nebo Dropbox.

Využití externího úložiště přináší spíše omezení než výhody. Snad jedinou výhodou je to, že se externí úložiště stará o zálohování dat. Jako největší nevýhodu vidím to, že služba, která bude poskytovat dostatečně velké úložiště a dostatečnou rychlost přenosu dat, nebude bezplatná. Například služba Dropbox poskytuje zdarma úložiště velikosti 2 GB, což je značně nedostačující vzhledem k tomu, že soubory současné Fitušky mají celkovou velikost okolo 6 GB.

Zbývají tedy možnosti ukládat do databáze nebo do souborového systému serveru. Ukládání do databáze by znamenalo ukládat soubory jako datový typ BLOB (Binary Large Object). Názory na to, kterou možnost vybrat, se napříč diskuzemi s zkušenostmi programátorů různí. Argumenty, proč použít databázi, jsou použití indexů a rychlejší hledání a úzké svázání souborů a záznamů v databázi (například propojení identifikátorů souborů a diskuzních příspěvků). Argumenty proti použití databáze jsou zbytečné

---

<sup>7</sup>XHTML je technologie rozšiřující známé HTML a zde je použita pro tvoření grafické části klienta.

<sup>8</sup>Backing Beans jsou třídy jazyka Java, které poskytují své funkce a data stránkám XHTML.

<sup>9</sup>„Návrhový vzor Data Access Object zapouzdří přístup k datovému zdroji a spravuje připojení k němu. Datový zdroj může být například uložen v databázi.“[14]



zvětšování objemu dat a tím snížení výkonu databáze a to, že při načtení datového typu BLOB musí databáze číst z disku a proto je jednodušší číst z disku rovnou.

Při rozhodování, jaké řešení použít, jsem vyšel z práce „To BLOB or Not To BLOB: Large Object Storage in a Database or a Filesystem?“, která říká: „Studie ukazuje, že pokud jsou soubory průměrně větší než 1 MB, tak NTFS má jasnou výhodu nad SQL Serverem. Pokud jsou soubory menší než 256 KB, tak má jasnou výhodu databáze. Uvnitř tohoto rozsahu záleží na aktuální zátěži a na stáří záznamů v systému.“[11] Očekávám, že uživatelé budou k příspěvkům přidávat především obrázky, fotografie a soubory PDF, které často přesahují velikost 1 MB. Z tohoto důvodu volím řešení ukládat soubory do souborového systému serveru.

Struktura (obrázek 4.6) bude velmi jednoduchá, protože v systému se vyskytují pouze 2 skupiny souborů. První skupina jsou soubory patřící k příspěvkům nebo k zadání zkoušky. Tyto soubory budou uloženy v adresáři „posts“. Druhou skupinou jsou profilové obrázky uživatelů, které se ukládají do adresáře „icons“. Poslední položka ve struktuře je profilový obrázek „defaultProfileIcon.png“, který je použit, pokud si uživatel nenastavil žádný vlastní profilový obrázek.



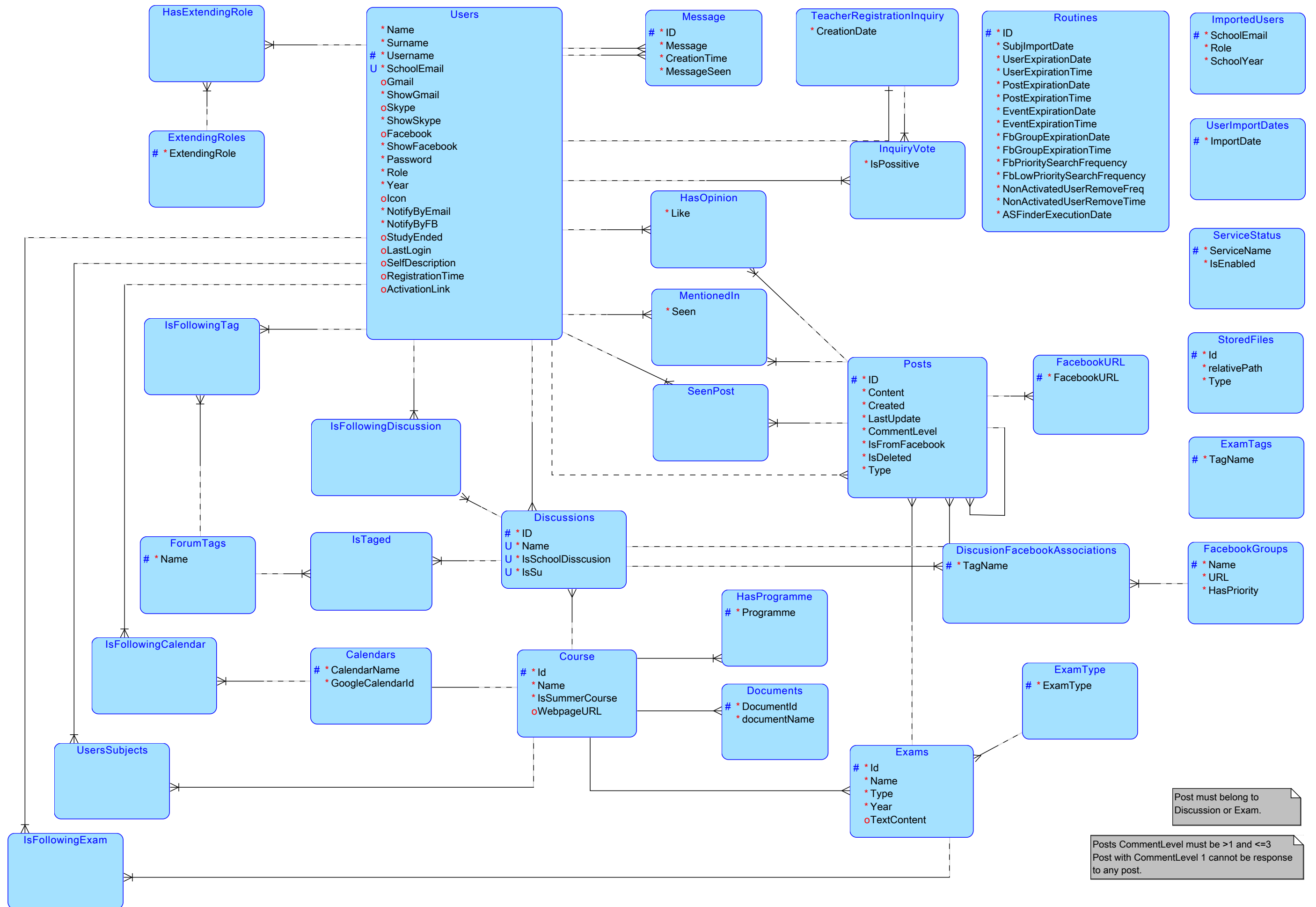
Obrázek 4.6: Struktura uložených dat v souborovém systému

Samotná databáze obsahuje všechna ostatní data mimo soubory, která systém potřebuje ke svému běhu a která do systému vkládají uživatelé. Podle toho, kdo data do databáze vkládá, lze schéma rozdělit na uživatelskou část a systémovou část.

Uživatelská část obsahuje především data, která vytvořili uživatelé, a pomocná data doplněná systémem (například datum vytvoření). Hlavní tabulky v této části jsou „Users“ (představuje registrované uživatele v systému), „Discussions“ (představuje všechny diskuze – předmětové, školní, mimoškolní a diskuze Studentské unie), „Course“ (obsahuje předměty, „Exams“ (obsahuje zkoušky patřící předmětům), „Message“ (obsahuje zprávy, které si uživatelé mohou zasílat), „Calendars“ (obsahuje Kalendáře Google, které byly vytvořeny skrze systém), „Documents“ (obsahuje Dokumenty Google, které byly vytvořeny skrze systém) a konečně tabulka „Posts“ (obsahuje všechny příspěvky). Další tabulky, které patří do této části, jsou nejčastěji vztahové tabulky nebo doplňující informace k výše uvedeným („ExtendingRoles“, „ExamType“, ...). Zajímavou tabulkou této části je „Posts“, která má vztah ke spoustě ostatních tabulek. Všechny tyto vztahy nelze popsat jazykem SQL, a proto musí být přidána integritní omezení, která musí být vynucená aplikací používající databázi. Prvním případem takového vztahu je příslušnost příspěvku buď diskusi nebo zkoušce, kdy příspěvek musí patřit právě jedné tabulce. Druhým případem je rekurzivní vztah příspěvku se svým rodičovským příspěvkem (lze vytvořit odpověď na příspěvek, čímž vznikne tento rodičovský vztah mezi původním pří-

spěvkem a odpovědí na něj). Každý příspěvek má nastavenou úroveň 1-3 a nesmí porušit pravidla, že rodič má úroveň o 1 nižší a příspěvek s úrovní 1 nesmí mít rodiče.

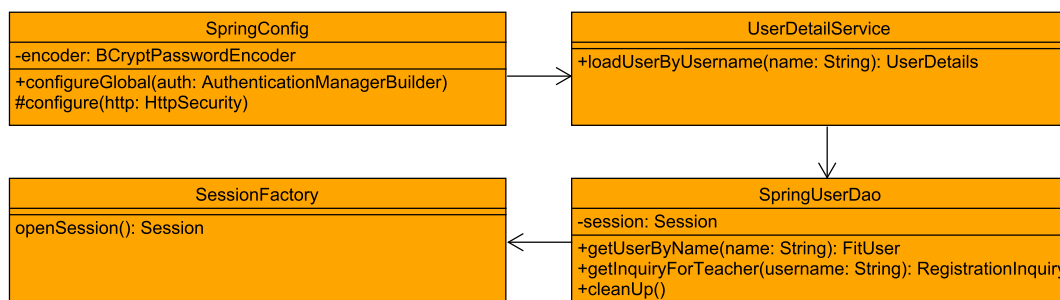
Systémová část obsahuje data týkající se nastavení samotného systému. Nejdůležitější tabulky jsou zde „Routines“ a „UserImportDates“, které obsahují nastavení spouštění rutin, a tabulka „ServiceStatus“, kde je uvedeno, jaká služba je zapnuta nebo vypnuta. Jak lze vidět na diagramu, tyto tabulky se vyznačují tím, že nemají žádné vazby na jiné tabulky.



## 4.7 Bezpečnost, přihlašování a Spring

Velmi důležitou částí systému s řízeným přístupem je samozřejmě autorizace, autentizace a celkově bezpečnost systému. Jako řešení těchto problémů vznikl framework Spring Security, který se zaměřuje na autorizaci a autentizaci aplikací v Java EE. Spring Security je součástí Spring Framework, který se zaměřuje na vývoj celé aplikace v Java EE. Pro použití Spring Security však není nutné použít celý Spring Framework a lze ho použít samotný.

Pro autentizaci v systému je využito klasické přihlášení uživatele pomocí uživatelského jména a hesla. Uživatel vloží informace do přihlašovacího formuláře, který je umístěn na speciální stránce, která je uvedena v nastavení v metodě *configure* třídy **SpringConfig** (obrázek 4.7). Spring z tohoto formuláře získá uživatelské jméno a heslo pokusí se získat údaje o uživateli metodou *loadUserByUsername* třídy **UserDetailsService**. Ta získá data z databáze pomocí třídy **SpringUserDao**, sestaví všechny informace a vrátí je třídě **SpringConfig**, která pak rozhodne, zda bude uživatel přihlášen. Pokud ano přidělí mu role získané z databáze. Spring dále rozeznává aktivitu uživatele pomocí HTTP relace<sup>10</sup>, a pokud se uživatel neodhlásí nebo nevyprší jeho relace, nemusí se znovu přihlašovat.



Obrázek 4.7: Diagram tříd při použití Spring Security

Přihlášením uživatele získá Spring Security informace o všech jeho rolích, pomocí kterých může provádět autorizaci (ověřování oprávnění uživatele). Ve třídě **SpringConfig** v diagramu 4.7 je uvedeno nastavení, které specifikuje, jaká role je nutná pro přístup k jaké webové stránce. Při požadavku na jakoukoliv stránku uvedenou v nastavení Spring Security zkontroluje role uživatele a provede jedno z následujících:

1. Uživatel má potřebou roli a přístup je povolen.
2. Uživatel nemá potřebnou roli a přístup je zamítnut.
3. Uživatel není přihlášen a je přesměrován na přihlašovací stránku.

<sup>10</sup> „Protokol HTTP nemá žádný stav a proto nelze rozpoznat, zda různé HTTP požadavky patří stejnému uživateli nebo ne, proto vznikla relace HTTP (anglicky HTTP session), která dokáže vytvářet kontext pro požadavky.“[22]

```

http.authorizeRequests().
    antMatchers("/admin/**").hasAnyAuthority("ADMINISTRATOR").
    antMatchers("/private/**").access("isAuthenticated()").
    antMatchers("/private/discussion.xhtml").hasAnyAuthority("STUDENT").

```

Obrázek 4.8: Ukázka nastavení omezení přístupu k webovým stránkám na základě rolí.

Poslední využití Spring Security v aplikaci je obrana proti útokům Cross-Site request forgery. „Cross-site Request Forgery (CSRF nebo také XSRF) je jedna z metod útoku na webové aplikací, pracující na bázi nezamýšleného požadavku pro vykonání určité akce v této aplikaci, který ovšem pochází z nelegitimního zdroje. Většinou se nejedná o útok směřující k získání přístupu do aplikace (i když i pro to může být zneužit) ale spíše využívá (zneužívá) akce uživatelů, kteří jsou k ní již v okamžiku útoku přihlášení.“[19] „Spring Security používá ochranu, která používá synchronizační řetězce. Zpracování každého požadavku vyžaduje tento řetězec, který je po přijmutí požadavku ověřen proti řetězcům, která má uložen server. Pokud se neshodují, tak požadavek selže.“[1] Tímto způsobem je aplikace proti CSRF útokům chráněna, ale je nutné, aby každý formulář obsahoval tento řetězec, jak je uvedeno v ukázce 4.9.

```

<form action="${logoutUrl}" method="post">
    <input type="submit" value="Log out" />
    <input type="hidden"
        name="${_csrf.parameterName}"
        value="${_csrf.token}"/>
</form>

```

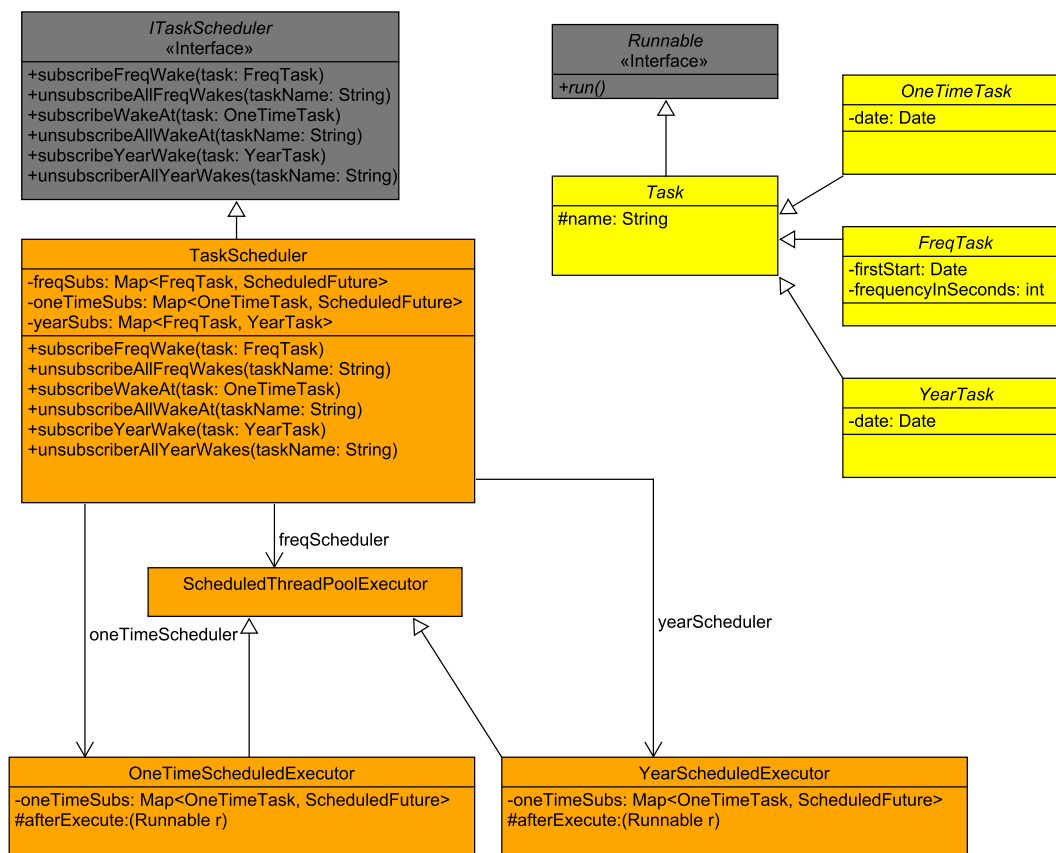
Obrázek 4.9: Ukázka formuláře se synchronizačním řetězcem.

## 4.8 Plánování úloh

Jednou z klíčových komponent aplikační vrstvy je plánovač úloh. Tato komponenta je nezbytná vzhledem k tomu, že systém potřebuje spouštět spoustu rutin v určitém čase nebo v určité dny. Příkladem rutiny, která plánovač využívá, je služba, která maže stará data a potřebuje být spuštěna každý rok v daný den. Systém spouští 3 typy rutin. První typ je rutina, která se spustí pouze jednou a to přesně v daný čas a den. Druhý typ rutiny se spouští přesně jednou za rok. Třetí typ rutiny se spustí poprvé v daný čas a potom se spouští opakovaně s danou frekvencí.

Prvním řešením návrhu plánovače bylo využít technologii EJB a plánované funkce. Plánovač by obsahoval metodu, která se provede každou minutu, zkontroluje seznam naplánovaných úloh a spustí ty, které mají být spuštěny. Efektivita tohoto řešení je nízká, protože metoda musí být spuštěna každou minutu i kdyby nebyla naplánovaná

žádná úloha a navíc by bylo nutné nasadit další technologii. Proto toto řešení nebylo použito.



Obrázek 4.10: Diagram tříd plánovače úloh

Jako daleko výhodnější možnost, jak plánovač navrhnout, je použití plánovače **ScheduledThreadPoolExecutor**, který je v jazyce Java již implementován. Tento plánovač zvládá opakovaně spouštět jednu rutinu s danou frekvencí a s daným zpožděním od naplánování a spustit rutinu jednou s daným zpožděním. Plánovač sice neumí spouštět rutinu, která se provádí opakovaně jednou za rok ale drobnou úpravou lze i toho s tímto plánovačem dosáhnout. Důvodem proč to nezvládne, i když umí spouštět rutinu opakovaně, je to, že frekvence opakování rutiny se zadává v milisekundách a 1 rok nelze vyjádřit přesně v milisekundách kvůli přestupným rokům.

Požadavky na plánovač jsou schopnost naplánovat všechny 3 typy rutin a toto naplánování kdykoliv zrušit. Tyto požadavky popisuje také rozhraní **ITaskScheduler** na diagramu 4.10. Typy rutin jsou reprezentovány třídami **OneTimeTask** (rutina, která se spustí pouze jednou), **FreqTask** (rutina, která se spouští periodicky) a **YearTask** (rutina, která se spouští jednou za rok). Všechny tyto třídy implementují rozhraní **Run-**

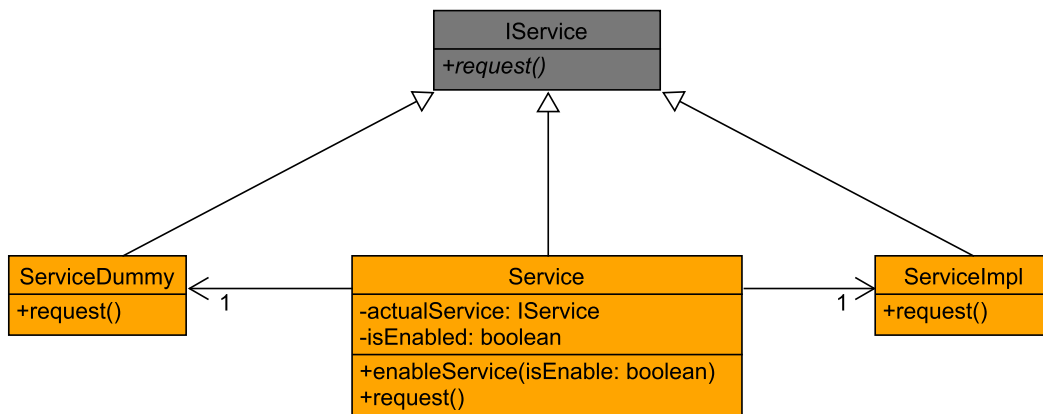
**nale**, takže je lze spustit jako samostatné vlákno, rozšiřují třídu **Task** a jsou abstraktní. Pro vytvoření rutiny je nutné vytvořit třídu, která rozšiřuje jednu ze tří tříd reprezentujících typy rutin, a musí implementovat metodu *run()*.

Samotný plánovač, který implementuje rozhraní **ITaskScheduler**, je reprezentován třídou **TaskScheduler**. Tato třída obsahuje 3 struktury pro uložení naplánovaných rutin a 3 spouštěče, které spouští každý jiný typ rutin. Rutina, která se vykonává periodicky, je plánována a spouštěna třídou **ScheduledThreadPoolExecutor**, která je již implementována v jazyce Java. Pro plánování rutiny, která se spustí pouze jednou, bylo použito stejnou třídu až na to, že po provedení musí být rutina smazána ze struktury *oneTimeSubs* a to tato třída neumí. Proto byla vytvořena třída **OneTimeScheduledExecutor**, která rozšiřuje třídu **ScheduledThreadPoolExecutor** a upravuje metodu, která se provede po provedení rutiny. Poslední rutinu má na starosti třída **YearScheduledExecutor**, která také rozšiřuje třídu **OneTimeScheduledExecutor** a upravuje metodu po provedení rutiny. Rutina je naplánována pouze na jedno spuštění a po jejím dokončení je v metodě *afterExecute()* spočítáno, kdy má být spuštěna znovu, a je naplánována znovu na jedno spuštění.

## 4.9 Služby s možností zapnutí a vypnutí

Většinu služeb lze vypnout a zapnout za běhu systému. Pokud služba přestane fungovat nebo není používána, může jí administrátor vypnout bez nutnosti restartu systému. Pro návrh těchto služeb byl využit návrhový vzor Strategie. „Návrhový vzor Strategie je užitečný pro situaci, kdy klientský objekt potřebuje dynamicky vybírat z množiny příbuzných algoritmů. Vzor udržuje implementace algoritmů v oddělených třídách a zapouzdřuje jejich volání.“[7]

Na diagramu (obrázek 4.11) je ukázán upravený návrhový vzor Strategie pro návrh služeb s možností zapnutí/vypnutí. Strategie, mezi kterými přepíná třída **Service**, jsou uloženy ve třídách **ServiceImpl** a **ServiceDummy**. Všechny 3 třídy implementují rozhraní **IService**, které specifikuje všechny metody služby (zde pro ukázkou specifikuje pouze metodu *request()*). Třída **ServiceImpl** reprezentuje strategii, kdy je služba zapnuta, a třída **ServiceDummy** reprezentuje strategii vypnuté služby. Třída **Service** deleguje volání všech metod specifikovaných rozhraním **IService** na jednu ze tříd na základě vnitřního stavu *isEnabled*, který lze měnit metodou *enableService()*.



Obrázek 4.11: Upravený návrhový vzor Strategie

## 4.10 Služby Google

Součástí systému je podle požadavků používání Kalendářů Google a Dokumentů Google. Tyto služby jsou dostupné skrze knihovny, které poskytuje sama společnost Google. Jedná se o knihovny `google-api-client`, `google-api-services-calendar` a `google-api-services-drive`. Pro používání služeb společnosti Google je nutné vytvořit projekt a účet na webové stránce <https://console.developers.google.com/>. V nastavení projektu se musí povolit služby Kalendáře Google a Dokumenty Google a vygenerovat soubor s přístupovými informacemi. Služby Kalendář Google a Dokumenty Google jsou v systému navrženy téměř identicky, proto dále popíši jen službu Kalendáře Google.

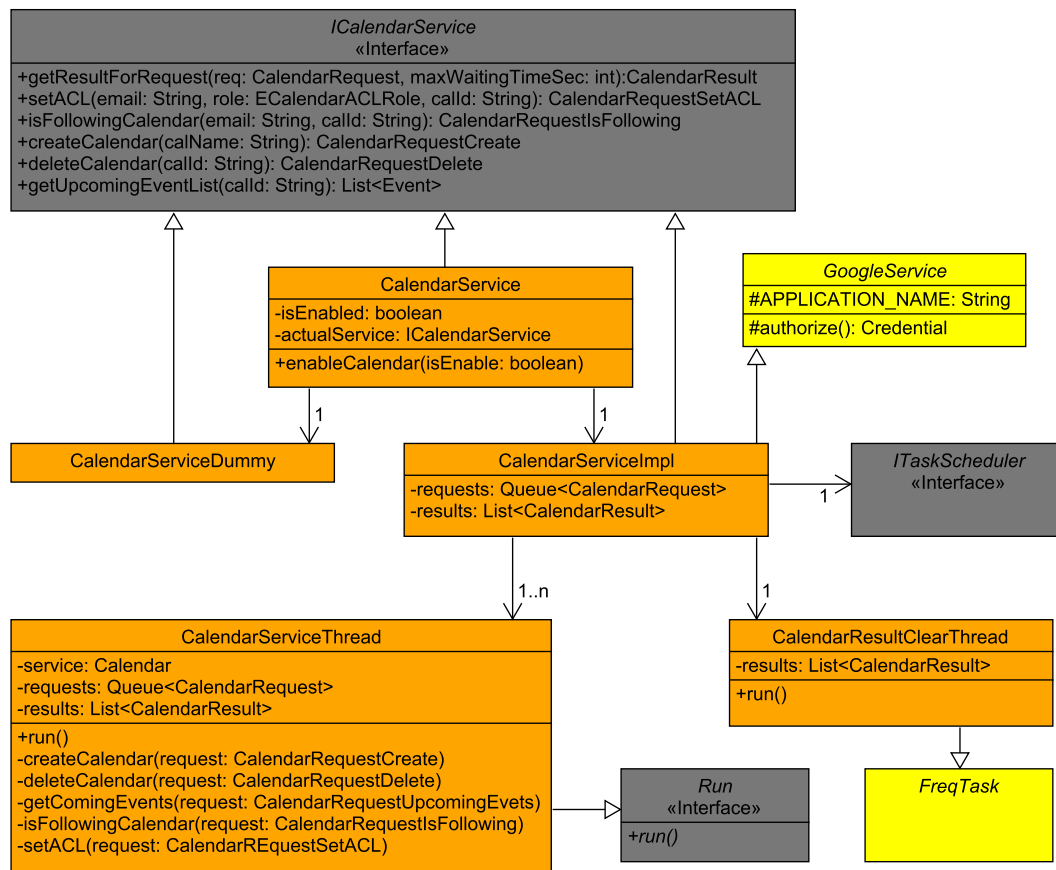
Služba Kalendáře Google je navržena podle vzoru, který je popsán v kapitole 4.9, proto je možné ji vypnout a zapnout. Strategii, kdy je služba zapnuta, reprezentuje třída **CalendarServiceImpl** v diagramu 4.12. Z důvodu velkého množství požadavků a delší doby zpracování požadavků (převážně na straně společnosti Google) jsem použil návrhový vzor Producent-konzument<sup>11</sup>, který rozloží zátěž mezi více vláken aplikace. V tomto případě je producent požadavků třída **CalendarServiceImpl**, která vkládá požadavky typu **CalendarRequest** do sdílené fronty `requests`. Konzumentem je třída **CalendarServiceThread**, která v nekonečné smyčce vybere požadavek z fronty, požadavek vyřídí a výsledek uloží do sdíleného seznamu `results`. Instance konzumenta se spouští v novém vlákně a je jich typicky více než 1.

V seznamu `results` jsou výsledky požadavků uloženy, dokud si je zadavatel „nevyzvedne“ metodou `getResultForRequest()`. Tento koncept může potenciálně způsobit nekonečný růst seznamu výsledků kvůli nevyzvednutým výsledkům. Proto má třída **CalendarServiceImpl** k dispozici rutinu **CalendarResultClearThread**. Tato rutina je

<sup>11</sup>„Producent-konzument je klasický příklad synchronizační úlohy. Problém popisuje 2 procesy, producenta a konzumenta, kteří sdílí fontu, do které producent vkládá data a konzument je odtud vybírá a zpracovává.“[21]

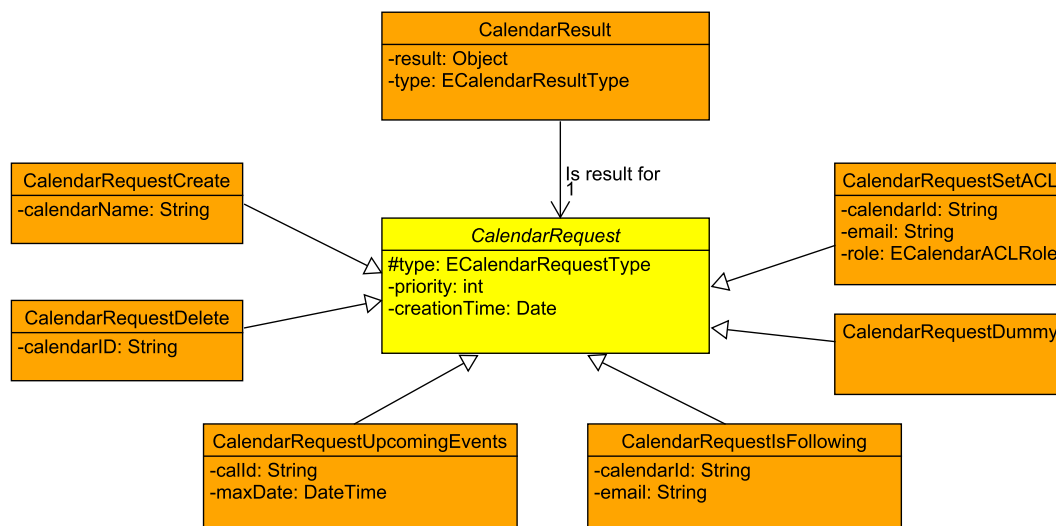


spouštěna plánovačem, který implementuje **ITaksScheduler**, a má za úkol při svém spuštění projít seznam results a smazat výsledky, které jsou již moc staré.



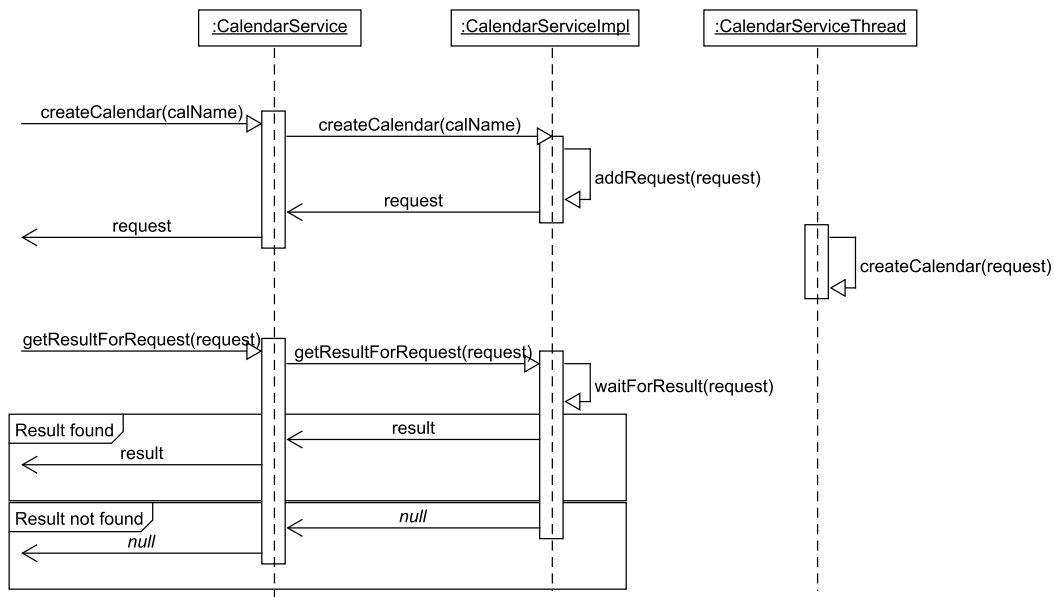
Obrázek 4.12: Diagram tříd služby Google Kalendář

Pro každý požadavek na službu Google Kalendáře je vytvořena třída, která obsahuje všechny potřebné informace pro vyřízení požadavku. Jmenovitě jsou to třídy **CalendarRequestCreate**, **CalendarRequestDelete**, **CalendarRequestUpcomingEvents**, **CalendarRequestIsFollowing** a **CalendarRequestSetACL** (diagram 4.13). Navíc vznikl požadavek **CalendarRequestDummy**, který když konzument vybere z fronty, tak ukončí své vlákno. Tento mechanismus slouží pro řádné ukončení systému při jeho vypnutí. Výsledek požadavku je reprezentován třídou **CalendarResult**, která obsahuje typ výsledku (úspěch, chyba, služba vypnuta nebo vyčerpán limit požadavků) a výsledek result typu **Object**. Protože každý objekt v jazyce Java je specializací třídy **Object**, lze do proměnné result uložit jakýkoliv objekt.

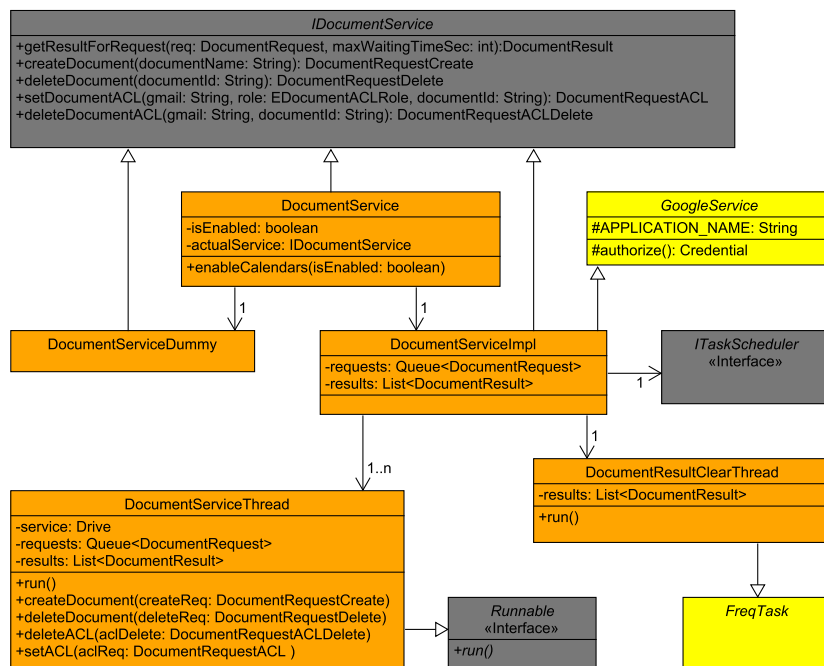


Obrázek 4.13: Diagram tříd požadavků a výsledku služby Google Kalendář

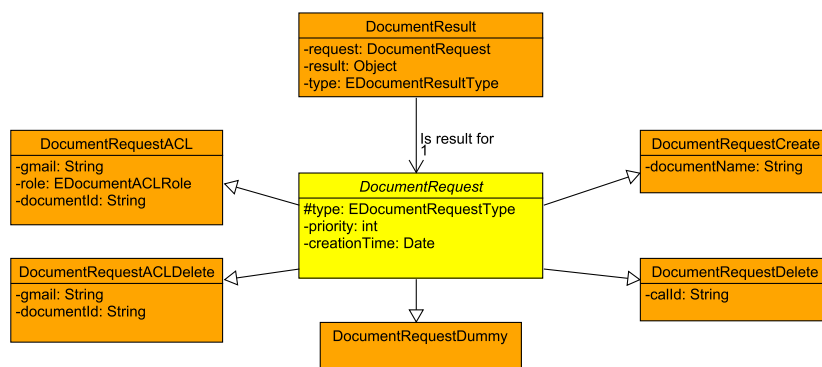
Vyřízení jednoho požadavku a typické použití služby Google Kalendáře, když je služba zapnuta, je ukázáno na diagramu sekvence 4.14. Klientský objekt zavolá metodu *createCalendar* třídy **CalendarService**. Protože je služba povolena, **CalendarService** deleguje vyřízení požadavku na třídu **CalendarServiceImpl**, která sestaví požadavek a vloží o do fronty. Vlákno **CalendarServiceThread** vybere požadavek z fronty, vyřídí ho a vloží výsledek do seznamu výsledků. Klientská třída zažádá o výsledek metodou *getResultForRequest*, která je opět delegována na **CalendarServiceImpl**. **CalendarServiceImpl** prohledá seznam výsledků a pokud není nalezen, tak na výsledek bude čekat. Pokud výsledek není přidán do seznamu než vyprší čas pro maximální čekací dobu, tak je vrácena hodnota **null**.



Obrázek 4.14: Diagram sekvence vyřízení požadavku službou Google Kalendář



Obrázek 4.15: Diagram tříd služby Google Dokumenty



Obrázek 4.16: Diagram tříd požadavků a výsledku služby Google Dokumenty

## 4.11 Služba Facebook

Služba Facebook je určena k integraci sociální sítě Facebook do systému. Má za úkol nahrávat příspěvky, které byly vloženy do Fitušky2, na síť Facebook a příspěvky ze sítě Facebook stahovat do Fitušky2 a třídit je. Systém bude prohledávat zadané skupiny v síti Facebook a bude hledat příspěvky, které jsou označeny značkami v hranatých závorkách. Pokud budou tyto značky asociované s některou diskuzí ve Fitušce2, bude příspěvek stažen a přidán do patřičné diskuze. Skupiny budou rozděleny na prioritní a neprioritní, kdy každý typ bude mít nastaven jiný interval prohledávání. V příkladu příspěvku se značkami (4.17) lze najít značky PES (zkratka předmětu), [3. úloha] a [2. příklad]. S těmito značkami a správně nastavenými asociacemi by byl příspěvek přidán do diskuze „3. úloha“ předmětu „PES - Petriho síť“.

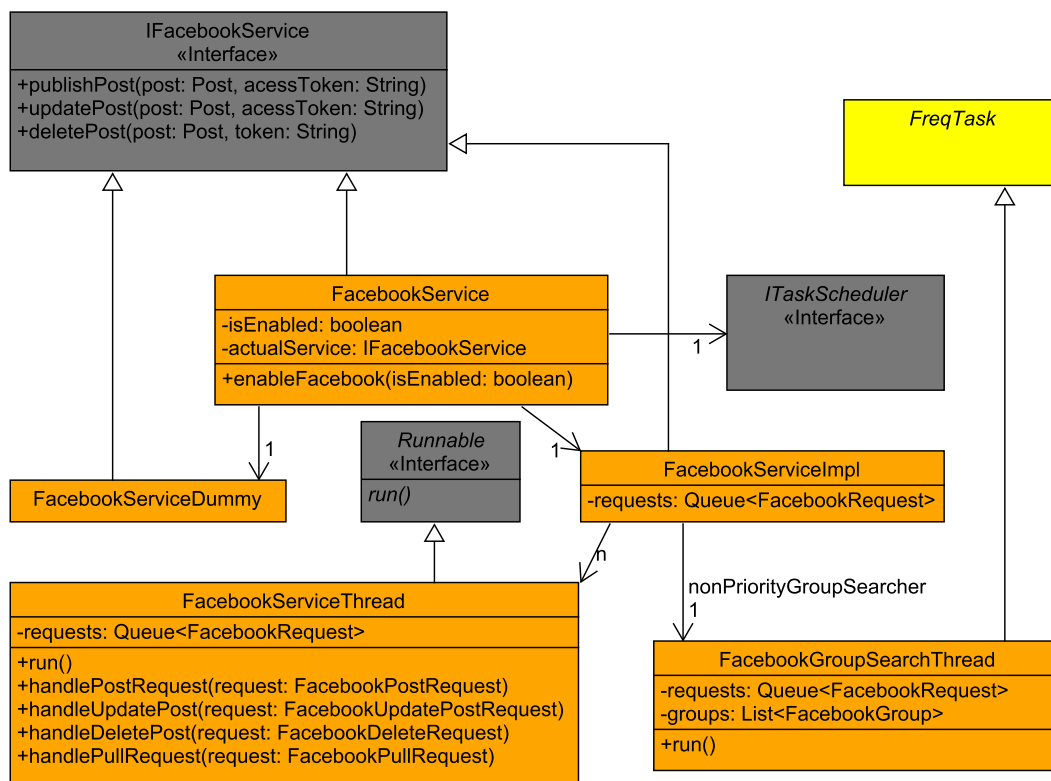
[PES] [3. úloha] [2. příklad] V grafe som našiel 11 cyklov, podľa prednášiek, ale nedarí sa mi nájsť počiatočné ohodnotenie siete, tak aby splňovala podminku bezpečnosti. Je možné, že príklad nemá riešenie pre bezpečnosť, alebo som úplne vedľa z tých cyklov? Díky

Obrázek 4.17: Ukázka příspěvku se značkami

Protože společnost Facebook neposkytuje vlastní knihovnu pro komunikaci s jejími sociální sítí pro jazyk Java, použil jsem knihovnu restFb, která má všechnu potřebnou funkčnost. Komunikace mezi sítí a knihovnou je prováděna pomocí GraphAPI, což je primární protokol pro komunikaci se sítí Facebook. Pro každý požadavek je potřeba mít speciální řetězec znaků, který vygenerovala síť Facebook a který dává jeho vlastníkovu oprávnění k omezené množině akcí. Tento řetězec se nazývá token a je ho hned několik druhů. Aplikace bude využívat pouze 2 typy těchto řetězců. První typ je token aplikace, který jeho držitele opravňuje k přístupu k datům Facebook aplikace, která byla vytvořena

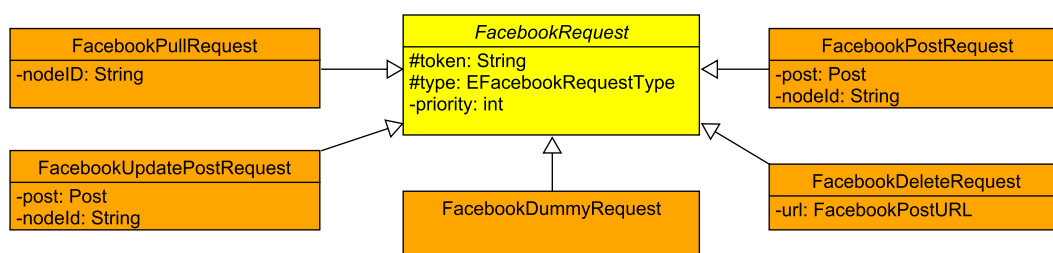
na webové stránce <https://developers.facebook.com/>. Tento řetězec bude použit pro prohledávání skupin, do kterých bude muset být účet aplikace přidán. Druhý typ řetězců je uživatelský token. Tento řetězec Fituška2 získá od uživatele skrze webové rozhraní. Uživatelský token bude použit pro vkládání příspěvku na síť Facebook pod Facebook účtem uživatele. Tento typ řetězce má poměrně omezenou platnost a proto ho bude muset uživatel vkládat opakovaně po jeho expiraci.

Návrh samotné služby Facebook v systému je založen na vzoru pro službu s možností zapnutí/vypnutí popsané v kapitole 4.9. Vypnutou službu v diagramu 4.18 reprezentuje třída **FacebookServiceDummy** a zapnutou službu představuje **FacebookServiceImpl**. Požadavky na službu jsou definované rozhraním **IFacebookService**, které obě zmíněné třídy implementují. Zpracovávání požadavků na zapnutou službu je realizováno návrhovým vzorem Producent-konzument, který umožňuje vyřizování požadavků a vkládání nových požadavků více vláken a tak zvýšení efektivity. Konzumentem je v tomto případě třída **FacebookServiceThread**, která v nekonečné smyčce vybírá požadavky z fronty requests a zpracovává je. Instance této třídy jsou spouštěny v novém vlákne a typicky bude instancí v systému více než 1. Producenty jsou třídy **FacebookService** a **FacebookGroupSearchThread**. Třída **FacebookService** vkládá do fronty požadavky, které přichází od uživatelů Fitušky. 2 instance **FacebookGroupSearchThread** (1 instance pro prioritní prohledávání skupin a 1 instance pro neprioritní prohledávání) vkládají do fronty požadavky, které vzniknou při prohledávání Facebook skupin. Obě instance jsou spouštěny plánovačem **ITaskScheduler**.



Obrázek 4.18: Diagram tříd požadavků služby Facebook

Požadavky na službu Facebook jsou následující – nahrát nový příspěvek na síť Facebook, aktualizovat již nahraný příspěvek, smazat příspěvek a prohlédnout Facebook skupinu. Pro každý tento požadavek je vytvořena třída, která obsahuje data nutná k jeho vykonání. Diagram 4.19 ukazuje tyto třídy, které všechny rozšiřují abstraktní třídu **FacebookRequest**. Navíc je v systému třída **FacebookDummyRequest**, která je použita pro signalizaci konzumentovi, že se má ukončit.



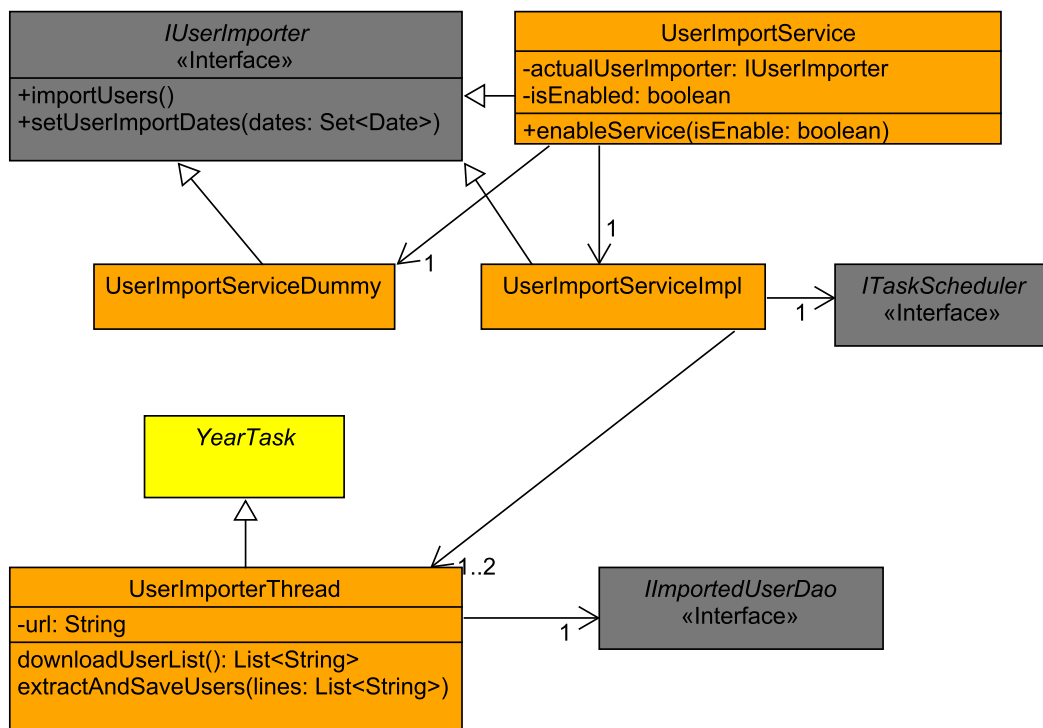
Obrázek 4.19: Diagram tříd požadavků služby Facebook

## 4.12 Služby pro importování uživatelů a předmětů

Protože systém obsahuje uživatele a předměty, což jsou data, která se každý rok mohou měnit, jsou vytvořeny služby pro import těchto dat ze školních systémů. Obě služby jsou vytvořeny podle vzoru pro návrh služby s možností zapnout/vypnout popsaném v kapitole 4.9 a obsahují rutiny, které lze naplánovat nebo spustit na vyžádání.

Služba pro importování uživatelů se skládá z tříd nutných pro vytvoření služby s možností zapnutí/vypnutí a třídy, která se spouští v novém vlákne a vykonává samotné importování. Tato třída se nazývá **UserImportThread** a jak je vidět na diagramu 4.20, tak rozšiřuje třídu **YearTask**, kterou lze naplánovat na spuštění jednou za rok. Jedna instance této třídy se spouští automaticky podle naplánování a druhá instance se vytvoří pokud je proveden požadavek na okamžité importování.

Samotné importování probíhá tak, že třída **UserImportThread** metodou *downloadUserList()* z nastaveného místa získá seznam studentů a uživatelů ve formátu „uživatelskéjméno,jméno,příjmení,zařazení“, kde zařazení je buď zkratka katedry u učitele nebo program a ročník u studenta. Z tohoto seznamu jsou vytrženy účty, které nelze zařadit mezi studenty nebo učitele, a následně jsou zbylé účty nahrány do databáze třídou implementující rozhraní **IImportedUserDao**. Studenti se v seznamu poznají tak, že jejich zařazení patří do jazyka následujícího regulárního výrazu, „FIT (BIT|M\w\w|D\w\w\w)(\d)r.\*“. Tímto výrazem je nalezen program (BIT, DVI4, MIS, MBS, MBI, ...) a ročník. Účet je identifikován jako učitelský, pokud jeho zařazení je řetězec 2-5 velkých písmen (odpovídají zkratce katedry).

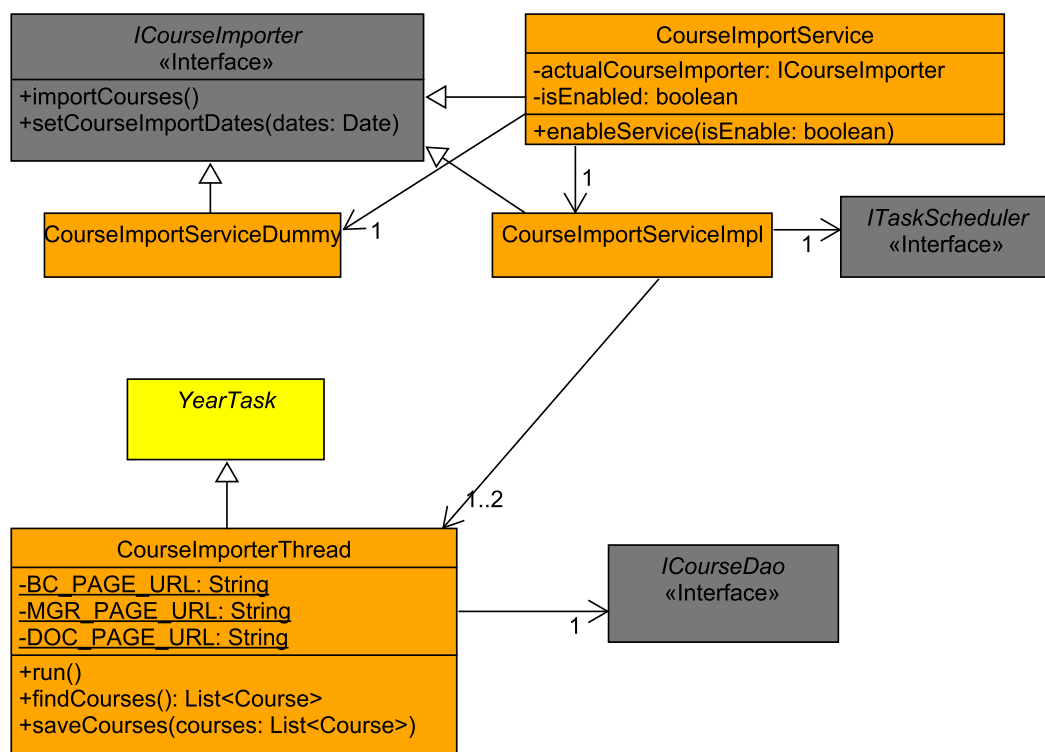


Obrázek 4.20: Diagram tříd služby pro import uživatelů

Služba pro importování předmětů je opět navrhnutá podle vzoru pro službu s možností zapnutí/vypnutí. Návrh této služby je zobrazen diagramem 4.21. Vykonavatelem importu je třída **CourseImporterThread**, která se je rozšířením třídy **YearTask** a je ji tedy možné plánovat pro spuštění jednou ročně. Třída **CourseImportServiceImpl**, která reprezentuje zapnutou službu, používá 2 instance třídy **CourseImporterThread**. První instance je plánována třídou implementující **ITaskScheduler** pro spuštění jednou ročně a druhá instance se vytvoří při vyžádání okamžitého provedení importu předmětů.

Protože seznam předmětů nelze nikde získat ve formě vhodné pro počítačové zpracování, musí hledání předmětů probíhat dolováním ze školních webových stránek. Na těchto stránkách se hledá tabulka s identifikátorem „#contenttable“. Tato tabulka se následně prochází řádek po řádku a hledají se řádky, které odpovídají regulárnímu výrazu „(\w{3,5}) (V | P | PVT | PVA | PVN | PVC | PVF | PVH | PVS) (\d{1,2}) (.\*)“. První skupina, kterou výraz nalezne, obsahuje zkratku předmětu, druhá skupina obsahuje program předmětu a třetí skupina obsahuje počet kreditů. Sloupec za touto trojicí obsahuje název předmětů. S touto čtveřicí je možné již předmět vytvořit a uložit do seznamu. Protože je možné, aby 1 předmět byl ve více programech, tak po tom, co byl nalezen předmět v tabulce, který v seznamu již je, tak se pouze přidá program předmětu v tabulce.





Obrázek 4.21: Diagram tříd služby pro import předmětů

## 4.13 Ostatní služby

### Služba $\text{\LaTeX}$

Pomocí editoru může uživatel vložit do příspěvku matematický vzorec v jazyce  $\text{\LaTeX}$ . Proto je nutné navrhnout službu, která pro vzorec v jazyce  $\text{\LaTeX}$  vytvoří obrázek tohoto vzorce, který bude v příspěvku zobrazen. Služba je implementována pomocí technologie Servlet<sup>12</sup>, která zpracovává požadavky na nastavené webové adrese. Parametr požadavku je kód v jazyce  $\text{\LaTeX}$ , ze kterého má být vytvořen obrázek.

Servlet například pro požadavek

`www.Fituska2.cz/latexImages?%5Cprod_%7Bi%3Da%7D%5E%7Bb%7D%20f%28i%29`

který obsahuje zakódovaný parametr „ $\text{\prod}_{i=a}^b f(i)$ “ vrátí obrázek 4.22.

<sup>12</sup> „Servlet je třída v jazyce Java určená pro rozšíření možností serveru pomocí modelu požadavek-odpověď. I když Servlet může odpovědět na jakýkoliv typ požadavku, většinou se používá pro rozšíření aplikací na webových serverech. HTTP Servlet dokáže zpracovávat HTTP požadavky typu GET a POST a odpovídat na ně.“[6]

$$\prod_{i=a}^b f(i)$$

Obrázek 4.22: Vytvořený obrázek pro vzorec v jazyce L<sup>A</sup>T<sub>E</sub>X

## Služba odstraňující stará data

Protože data do databáze budou neustále přibývat, je nutné nějaká data i odstranit, aby databáze nerostla do nekonečna. Jako u většiny databází podobného typu je i v této stanoven limit jak stará data se mohou v databázi nacházet. Toto omezení se týká převážně diskuzních příspěvků a uživatelských účtů. Mazání uživatelských účtů má nejen význam pro omezení růstu velikosti databáze ale i pro zamezení přístupu uživatelům, kteří již delší dobu nejsou studenty nebo učiteli.

Mazání starých příspěvků i uživatelských účtů má na starosti služba **OldDataRemoveService**, která je navrhnutá podle vzoru popsaném v kapitole 4.9, tak aby bylo možné vypnout automatické mazání uživatelů a příspěvků. Nastavení okamžiku mazání a maximální stáří příspěvků a účtů je možné přes webové uživatelské rozhraní.

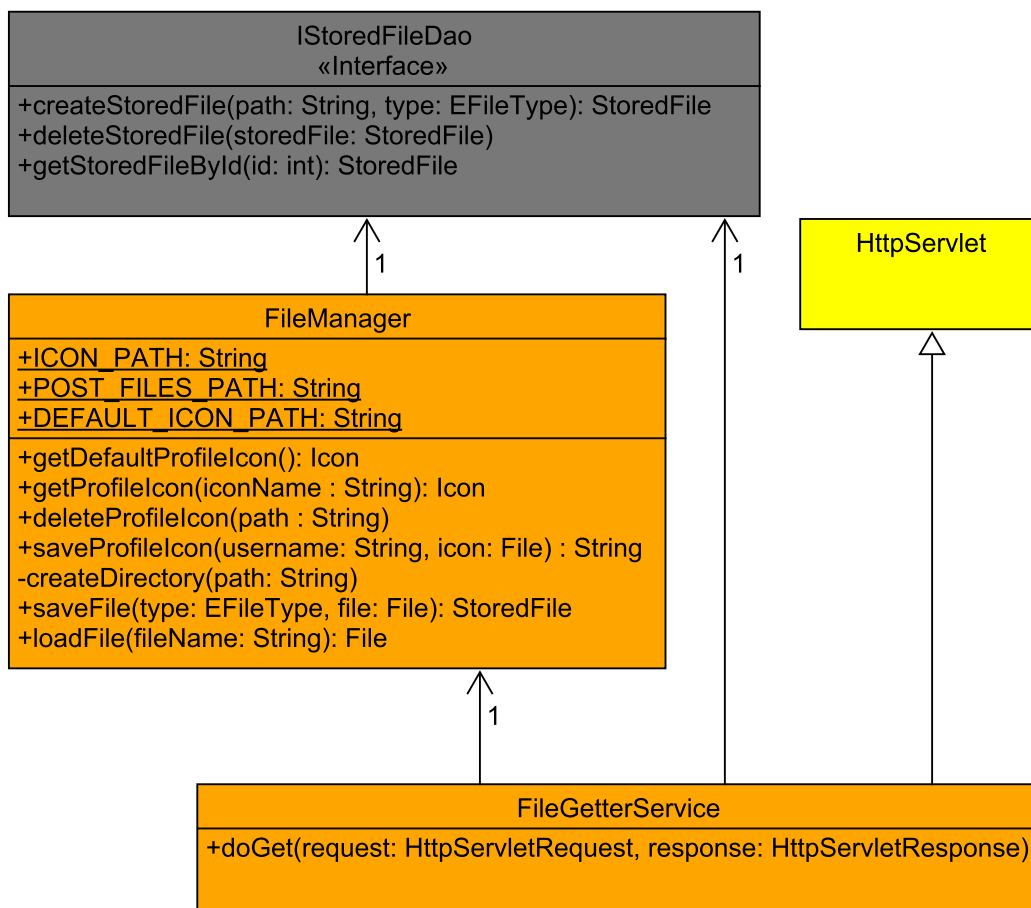
## Služba pro ukládání souborů a jejich zpřístupnění

Protože soubory vložené do příspěvků a profilové obrázky uživatelů se ukládají do souborového systému serveru, je nutné navrhnout službu pro ukládání a zpřístupnění těchto souborů. Návrh této služby popisuje diagram 4.23. Služba se skládá ze dvou tříd, **FileManager** a **FileGetterService**.

Třída **FileManager** slouží pro ukládání a mazání souborů vložených do příspěvků a pro ukládání, mazání a získání profilových obrázků. Pokud služba přijme požadavek na získání profilového obrázku uživatele, který si obrázek nenastavil, tak služba vrátí základní obrázek, který se nachází na adrese uložené v atributu `DEFAULT_ICON_PATH`. Profilové obrázky jsou uloženy do adresáře s adresou uloženou v atributu `ICON_PATH` a soubory k příspěvkům jsou uloženy do adresáře, jehož adresa je uložena v atributu `POST_FILES_PATH`. Informace o uložených souborech jsou uloženy v databázi pomocí třídy implementující rozhraní **IStoredFileDao**.

Jak lze na diagramu vidět, tak metoda *saveFile()* nevrací uložený soubor ale třídu obsahující informace o tomto souboru. Z těchto informací se v klientské třídě, která soubor pomocí třídy **FileManager** uložila, vytvoří webový odkaz, kde je tento soubor dostupný. K zajištění dostupnosti těchto souborů je vytvořena třída **FileGetterService**. Tato třída rozšiřuje abstraktní třídu **HttpServlet**, která z této třídy dělá službu s možností vyřizování HTTP požadavků na předem nastavené webové adrese. Pokud tyto požadavky obsahují parametr `id`, třída tento soubor vyhledá v databázi pomocí třídy implementující rozhraní **IStoredFileDao**, soubor otevře a odešle ho v odpovědi na tento požadavek.

Ukázka požadavku na třídu **FileGetterService**  
[www.Fituska2.cz/files?id=31](http://www.Fituska2.cz/files?id=31)



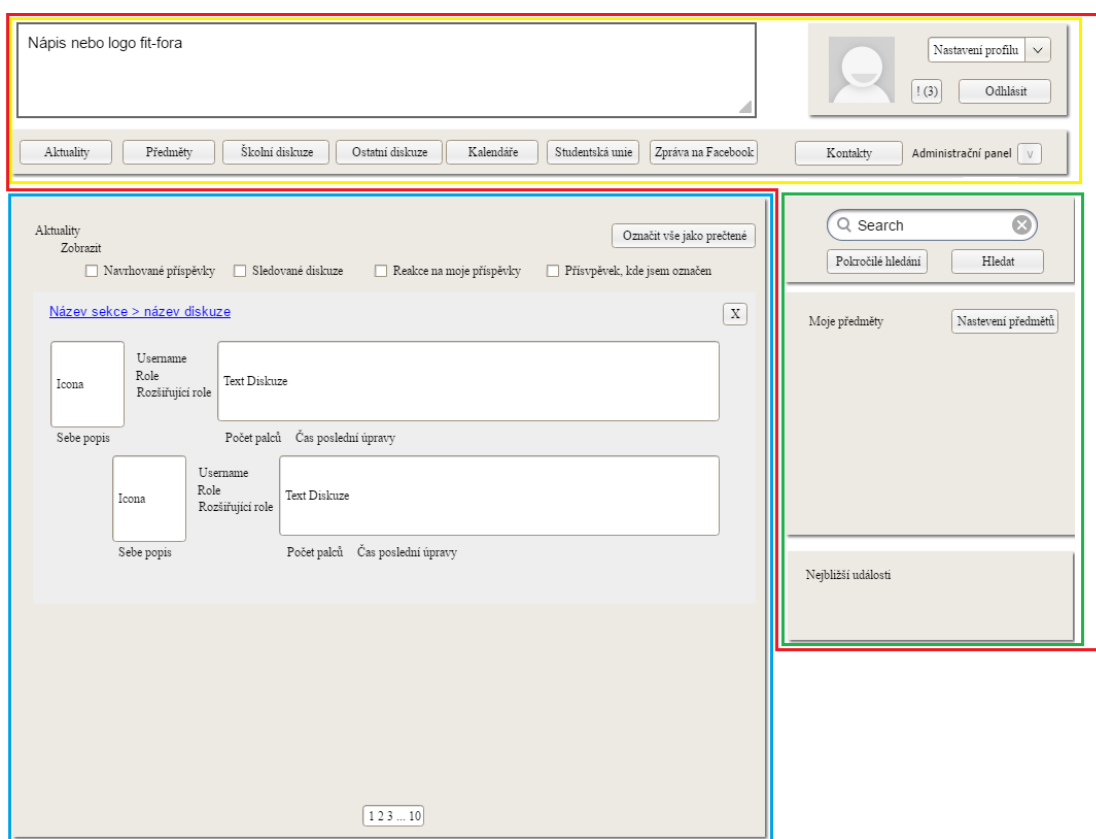
Obrázek 4.23: Diagram tříd služby pro ukládání a zpřístupnění souborů

## 4.14 Uživatelské rozhraní

Uživatelské rozhraní systému bude implementováno v podobě webových stránek v jazyce XHTML, kdy bude použit framework Primefaces. Při návrhu stránek jsem se snažil využít prostor při zobrazení na širokoúhlém monitoru.

Každá stránka je složena ze šablony, která je na většině stránek stejná, a z vlastního obsahu stránky. Například na stránce, která zobrazuje aktuality (obrázek 4.24), jsou tyto části vidět. Celá šablona je ohraničená červenou čarou a skládá se z horního panelu (uvnitř žlutého obdélníku) a pravého bočního panelu (uvnitř zeleného obdélníku). Hlavní část stránky je ohraničena modrou barvou. Horní panel šablony je použit pro navigaci systému (ta je pro všechny uživatele se stejnou rolí stejná) a pro zobrazení okénka uživatele, kde se uživatel může odhlásit, přejít na nastavení svého profilu nebo kde může vidět počet nepřečtených zpráv. Pravý boční panel slouží k vyhledávání a zobrazení

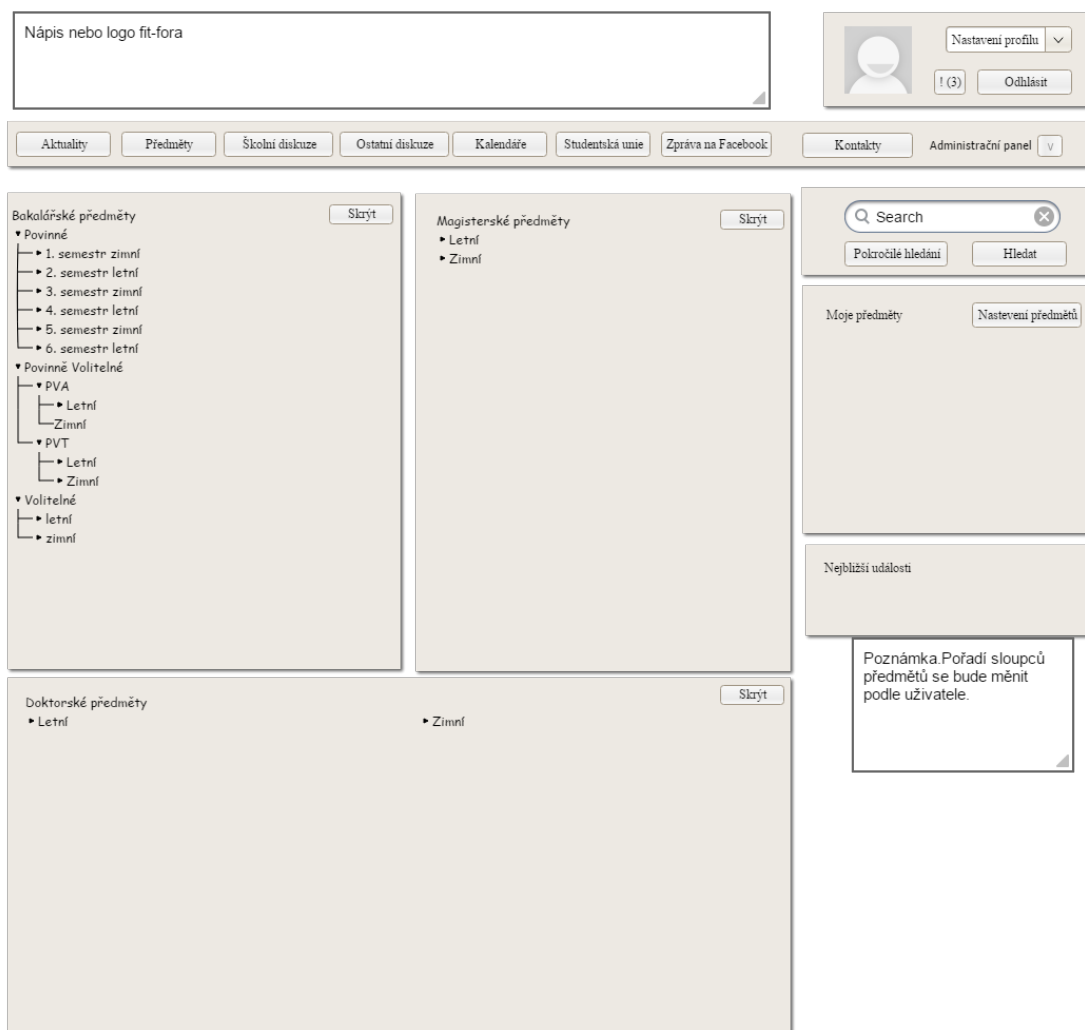
předmětů, které má uživatel zapsány a nejbližších událostí z kalendářů, které uživatel sleduje. Hlavní část stránky zobrazuje příspěvky, které jsou uživateli doporučeny.



Obrázek 4.24: Návrh stránky s aktualitami

V této kapitole jsou popsány a umístěny návrhy vybraných webových stránek. Obrazky všech návrhů lze najít v příloze C.

První návrh se týká zobrazení všech předmětů (obrázek 4.25). Předměty jsou na první úrovni rozděleny na bakalářské, magisterské a doktorské. Magisterské a doktorské předměty nelze dělit dál než na letní a zimní. Doktorské předměty žádné jiné dělení nemají mimo dělení na povinné a volitelné, kdy povinné předměty obsahují pouze předmět „Zkouška z jazyka anglického pro Ph.D.“. Z tohoto důvodu nejsou dál rozděleny. U magisterských předmětů dochází k opačnému problému, protože je lze dělit na „Povinné“, „Volitelné“ a „Povinně volitelné“, kdy každý magisterský studijní program dělí předměty jinak. Proto jediné složitější dělení je vytvořeno pouze u bakalářských předmětů.



Obrázek 4.25: Návrh stránky všech předmětů

V analýze současného systému jsem kritizoval dělení diskuzí nejprve podle roku vytvoření a až potom podle předmětu. Na obrázku 4.26 je tedy vidět rozdělení, které je daleko přehlednější. Stránka zobrazuje detail jednoho předmětu, který může obsahovat diskuze, diskuze týkající se zkoušek a seznam odkazů na dokumenty Google. Diskuze samotné podle roku nejsou členěny vůbec, protože jejich obsah nemusí být vůbec závislý na roku. Naopak zkoušky jsou členěny jak podle roku, tak i podle typu zkoušky. Dokumenty jsou zobrazeny také bez členění, protože ze zkušenosti s dokumenty, které se předávají přes sociální sítě, vím, že se dokumenty častěji vyvíjejí a aktualizují namísto vytváření nových dokumentů pro každý ročník.

Pokud má uživatel dostatečné oprávnění (má rozšiřující roli *moderátor* a vyšší nebo je autorem diskuze nebo zkoušky), může diskuzi a zkoušku editovat a smazat dokument.

Pokud toto oprávnění má, na stránce se zobrazí příslušná tlačítka, pokud ne tlačítka se nezobrazí. Stejně pravidlo platí i pro zobrazení celé sekce zkoušky a dokumenty. Pokud uživatel nemá základní roli *student* nebo *učitel* s rozšiřující rolí *Přístup ke zkouškám*, tak se mu tyto sekce vůbec nezobrazí.

Obrázek 4.26: Návrh stránky detailu předmětu

Každý uživatel má právo vytvářet diskuze v sekci, ke které má přístup. Jediné omezení, které pro vytváření diskuze platí je, že nesmí být 2 diskuze se stejným jménem ve stejné sekci. Návrh stránky, na které se vytváří diskuze, je zobrazen na obrázku 4.27. Pro vytvoření diskuze je nutné jen zadat jméno, ale je možné jí přiřadit fórum značky a vytvořit asociaci se skupinou na síti Facebook. Pro výběr fórum značek a skupin sítě Facebook je použita komponenta *PickList*, která má 2 seznamy reprezentující vybrané a nevybrané dostupné položky a tlačítka pro přesun položek mezi seznamy. Tato komponenta se hodí zejména pro výběr dat z kratšího seznamu, popřípadě i delšího, pokud by bylo přidáno vyhledávání, aby uživatel nemusel procházet desítky položek. Úspěšně přiřazené asociace jsou zobrazeny ve zvláštní tabulce pro větší přehlednost. Všechna tato nastavení lze změnit i po vytvoření diskuze pomocí editace diskuze, pokud má uživatel dostatečná práva (vlastník diskuze nebo má uživatel rozšiřující roli *moderátor* nebo vyšší).

Nápis nebo logo fit-fora

Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrací panel

Název nové diskuze

Text box

Vytvořit

Forum tagy diskuze

Přidané tagy

1BIT

2BIT

<

>

Dostupné tagy

1MIT

2MIT

3BIT

DVI

Vybrané asociace diskuze s facebook tagy

Název tagu	Název facebook skupiny	Název tagu	Název facebook skupiny
	Název facebook skupiny		Název facebook skupiny
	Název facebook skupiny		Název facebook skupiny
Název tagu	Název facebook skupiny	Název tagu	Název facebook skupiny
	Název facebook skupiny		Název facebook skupiny

Editace Přidání tagu

Název tagu

Text box

Přidané skupiny facebook k tagu

VUT FIT MIT 2016-2018 Teambuilding

VUT FIT MIT 2016-2018

Dostupné facebook skupiny

VUT FIT BIT 2015-2018

VUT FIT BIT 2014-2017

VUT FIT BIT 2013-2016

<

>

Přidat tag

Moje předměty

Nastavení předmětů

Nejbližší události

Q Search

Pokročile hledání

Hledat

Obrázek 4.27: Návrh stránky pro vytvoření diskuze

Návrh samotného zobrazení detailu diskuze a zobrazení příspěvků (obrázek 4.28) se drží obecně používaného návrhu, kdy jsou příspěvky zobrazeny pod sebou, odpovědi na příspěvky jsou přímo pod původním příspěvkem a jsou mírně odsazeny pro přehlednost. Samotný příspěvek má v levé části zobrazeny informace o autorovi (profilový obrázek, uživatelské jméno, základní role, rozšiřující role a sebe popis). Zobrazení všech rolí je velmi důležité, protože může přidávat na důležitosti odpovědi. Pokud na otázku týkající se předmětu odpoví učitel, který tento předmět vyučuje, lze tuto odpověď považovat za přesnou a důležitou. Stejně tak pokud administrátor odpovídá na otázku ohledně

51

běhu systému. Pravá strana příspěvku obsahuje text příspěvku, který je vnitřně uložen v jazyce HTML, čas poslední editace a hodnocení. Pokud má uživatel rozšiřující roli vyšší než *moderátor* nebo je autorem příspěvku, zobrazí se v této části i tlačítka na editaci a smazání příspěvku.

Zobrazení všech příspěvků je rozděleno na stránky, aby nedošlo k tomu, že se uživateli prohlížeč pokusí zobrazit stovky příspěvků, které se v diskuzi potenciálně mohou objevit. Samozřejmostí je, že se příspěvky řadí podle data vytvoření pro přehledné zobrazení.

Posledním prvkem stránky je tlačítko v horní části obsahové části stránky, které dovoluje diskuzi sledovat, nebo toto sledování zrušit. Pokud uživatel diskuzi sleduje, tak se mu budou nové příspěvky v diskuzi zobrazovat na stránce s aktualitami. Přehled všech sledovaných diskuzí, možnost zrušit sledování a přidat nové sledování je možné v nastavení uživatelského účtu uživatele.

Nápis nebo logo fit-fora

Nastavení profilu  
! (3) Odhlásit

Aktuality Předměty Školní diskuze Ostatní diskuze Kalendář Studentská unie Zpráva na Facebook Kontakty Administrační panel

Předmět Sledovat Zrušit sledování

Nový příspěvek

Diskuze

Icona Username Role Rozšiřující role Text Diskuze Del Edit

Sebe popis Odpověď Palec Up Palec Down Počet palců Čas poslední úpravy

Icona Username Role Rozšiřující role Text Diskuze Del Edit

Sebe popis Odpověď Palec Up Palec Down Počet palců Čas poslední úpravy

Search Pokročilé hledání Hledat

Moje předměty Nastavení předmětů

Nejbližší události

Poznámka. Facebook diskuze bude barevně odlišena.

1 2 3 4 ... 10

Obrázek 4.28: Návrh stránky diskuze



Návrh stránky zobrazující zkoušku, jak je vidět na obrázku 4.29, je velmi podobný stránce s detailem diskuze. Hlavní rozdílem je zobrazení zadání zkoušky, které se diskuze týká, a zobrazení dvou typů příspěvků. První typ je řešení, který obsahuje správné odpovědi na otázky v zadání nebo postup, který k výsledku vede. Druhý typ je klasický diskuzní příspěvek obsahující cokoliv až na řešení. Typy příspěvků jsou od sebe odlišeny a lze je skrýt/zobrazit tlačítky v horní části obsahové části stránky. Stejně tak lze skrýt i zadání. Jako v klasické diskuzi lze upravovat a mazat příspěvky s příslušným oprávněním. Upravit lze i zadání zkoušky, ale tuto akci může provést pouze uživatel s rozšiřující rolí vyšší než *moderátor*, protože autor zkoušky a zadání zkoušky se v databázi na požadavek Studentské unie neeviduje.

Nápis nebo logo fit-fora

Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrální panel

Předmět

Zobrazit diskuzi

Zobrazit řešení

Sledovat/Zrušit sledování

Zkouška

Del

Edit

Skrýt

PDF/IMG/TEXT

Q Search

Pokročilé hledání

Hledat

Moje předměty

Nastavení předmětů

Nejbližší události

Diskuze

Nové řešení

Nový příspěvek

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del

Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců

Čas poslední úpravy

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del

Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců

Čas poslední úpravy

Řešení

Nové řešení

Nový příspěvek

Icona

Username

Role

Rozšiřující role

Řešení

Del

Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Správnost řešení

Čas poslední úpravy

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del

Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců

Čas poslední úpravy

1 2 3 ... 10

Obrázek 4.29: Návrh stránky zobrazující zkoušku

54

## Kapitola 5

# Implementace klientské části

### 5.1 JSF a Primefaces

Implementace webových stránek v jazyce XHTML probíhala většinou podle provedených návrhů. Dále byly implementovány stránky, které pro svou jednoduchost navrženy nebyly. Stejně jako v návrhu je obsah stránek v XHTML složen ze šablony a obsahové části. Technologie JSF podporuje tento typ tvoření stránek v XHTML tak, že nejprve je vytvořena celá validní stránka v XHTML (obsahuje hlavičku, tělo, ...), která obsahuje vše co má obsahovat šablona, a potom se označí se místa, kam se bude vkládat další obsah při tvoření finální stránky pomocí této šablony. Následuje ukázka struktury šablony „MenuTemplate.xhtml“ použité pro tvoření stránek ve Fitušce2.

MenuTemplate.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:p="http://primefaces.org/ui"
      xmlns:ui="http://xmlns.jcp.org/jsf/facelets">
  <h:head>
    <meta http-equiv="Content-Type" content="text/html;
      charset=UTF-8" />
    <title>
      <ui:insert name="pageName">PageName</ui:insert>
    </title>
  </h:head>
  <h:body>
    <p:growl id="growl" life="4000"/>
    <p:layout id="mainLayout" fullPage="true">
      <p:layoutUnit position="north" size="160" >
```

```

        <p:layout style="height: 150px">
            <p:layoutUnit position="center">
                <h1>Fituška</h1>
                <h2>Testovací provoz</h2>
            </p:layoutUnit>
            <p:layoutUnit position="east" size="290">
                <!-- Panel s ikonou uživatele a navigací
                     pro uživatelské nastavení -->
            </p:layoutUnit>
            <p:layoutUnit position="south" size="45">
                <!-- Panel s hlavní navigací -->
            </p:layoutUnit>
        </p:layout>
    </p:layoutUnit>
    <p:layoutUnit position="center">
        <ui:insert name="content">
            <!-- Místo pro vložení obsahu -->
        </ui:insert>
    </p:layoutUnit>
    <p:layoutUnit id="LUeast" position="east" size="290">
        <!-- Pravý boční panel -->
    </p:layoutUnit>
</p:layout>
</h:body>
</html>

```

Šablona používá komponenty `<layout>` a `<layoutUnit>`, které poskytuje framework Primefaces pro základní rozvržení obsahu podle jejich pozice na obrazovce monitoru (sever, jih, východ a západ). Místa, kam se později vkládá další obsah při tvoření finální stránky, jsou označeny značkou `<insert>` a jejím parametrem `name`. V této šabloně je označené místo v hlavičce se jménem „pageName“, kam se vloží název stránky, a místo pro vložení obsahu se jménem „content“. Dále uvádím základní strukturu stránky „aktuality.xhtml“, která je do šablony vložena.

```

aktuality.xhtml
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:p="http://primefaces.org/ui"
      xmlns:ui="http://xmlns.jcp.org/jsf/facelets">
    <ui:composition template="MenuTemplate.xhtml">
        <ui:define name="pageName">Aktuality</ui:define>
    </ui:composition>
</html>

```

```

<ui:define name="content">
    <h1>Aktuality</h1>
    <!-- obsah stránky aktuality -->
</ui:define>
</ui:composition>
</html>

```

Tato stránka nemá ani hlavičku ani tělo stránky, ale celý její obsah je uzavřen do značky *composition*, jejíž parametr *template* specifikuje použití šablony „MenuTemplate.xhtml“. Uvnitř této značky jsou kusy kódu uzavřeny ve značky „define“, která definuje, kam se tento kód vloží do šablony. V tomto případě se textový řetězec „Aktuality“ vloží na místo v šabloně označené jménem „pageName“ a tím se změní název stránky. Hlavní obsah stránky „aktuality.xhtml“ se vloží na místo označené jménem „content“. Stejným způsobem jsou vytvořeny všechny stránky v systému.

Pro takto vytvořené stránky XHTML jsou vytvořeny třídy v jazyce Java, které poskytují data a funkce, které stránka může volat, tzv. Backing Beans. Tvoření těchto tříd se drží pravidla, že třída se použije maximálně na 1 stránce, mimo třídy, která obsahuje údaje o přihlášeném uživateli.

Tvoření instancí těchto Backing Beans má na starosti technologie CDI a stará se i o jejich životní cyklus, který je určen nastavením třídy. „Nastavení životnosti se provádí pomocí anotace třídy ve zdrojovém kódu. Existuje několik možností životnosti třídy. Anotace „@RequestScoped“ zaručí, že instance třídy zanikne po vyřízení 1 požadavku protokolu HTTP, „@SessionScoped“ zaručí, že instance existuje po celou dobu 1 sezení uživatele, instance třídy s anotací „@ApplicationScoped“ existují po celou dobu běhu aplikace a anotace „@Dependent“ zaručí, že instance bude existovat tak dlouho, jak dlouho bude existovat instance třídy, do které byla vložena.“[6]. V základních typech životnosti chybí ještě jeden typ, a to ten, kdy stránka existuje po dobu vyřízení 1 HTTP požadavku, ale je schopná s klientem dále komunikovat pomocí technologie AJAX<sup>1</sup>. Tohoto chování poskytuje třída s anotací „@ViewScoped“.

Kvůli tomu, že životnost většiny tříd Backing Beans je omezena právě na 1 požadavek je nutné řešit problém předávání informací mezi těmito třídami. Například třídě, která obsahuje data pro zobrazení editace příspěvku, je nutné, při kliknutí na tlačítko editace, předat identifikátor příspěvku. V systému jsou použity 2 různé metody pro předávání těchto parametrů.

První metoda je předávání pomocí parametrů zakódovaných v požadavku. Tento parametr je třídou zpracující požadavek přečten a uložen. Druhá možnost je uložit hodnotu parametru do struktury, která se váže k sezení uživatele. První metoda je použita u stránek, u který je možné, že uživatel bude chtít uložit nebo poslat odkaz na ně. Metoda není použita u všech stránek, protože by vynucovala dodatečnou autorizaci. Například

<sup>1</sup> „AJAX je technologie, která umožňuje prohlížeči poskytnout uživateli přirozenější prohlížení webových stránek. Před technologií AJAX vedla jakákoliv akce uživatele na odeslání požadavku a vykreslení celé webové stránky. Technologie AJAX umožňuje komunikaci mezi uživatelem a serverem pomocí asynchronní komunikace a aktualizaci pouze části webové stránky. Výsledkem je, že při požadavku se nevykresluje celá webová stránka ale pouze ta část, která se změnila po provedení uživatelské akce.“[12]

u stránky, která dovoluje upravit příspěvek a parametr by byl součástí webové adresy, bylo by možné upravit jakýkoliv příspěvek, u kterého známe jeho identifikátor. Proto u těch stránek, kde by uživatel pomocí změny parametru zakódovaného v požadavku mohl získat přístup k datům, ke kterým nemá potřebná oprávnění, je použita druhá metoda.

Příklad zakódování parametru `discId` do požadavku

`http://www.Fituska2.cz/private/discussionDetail.xhtml?discId=25750`

Příklad při zakódování parametru do struktury vázané na sezení uživatele

`http://www.Fituska2.cz/private/postEdit.xhtml`

Přečtení parametru v třídě vázané ke stránce `postEdit.xhtml`

```
Map<String, Object> sessionMap = FacesContext.getCurrentInstance().  
    getExternalContext().getSessionMap();  
post = (Post) sessionMap.get("postEdit_post");
```

## 5.2 Editor příspěvků

Klíčovou součástí systému je komponenta pro vytvoření nebo editaci příspěvku. Pro implementaci byl vybrán hotový editor CKEditor napsaný v jazyce JavaScript. Framework Primefaces obsahuje rozšíření Primefaces Extensions, které obsahuje tento editor upravený pro použití s technologií JSF. Samotný editor dokáže tvořit vše specifikované v požadavcích na obsah příspěvku až na vzorce v jazyce  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  a nahrání souborů. Editor byl vybrán protože je interaktivní, jednoduchý na použití a ukládá vytvořený obsah ve formátu HTML. Nahrávání souborů je realizováno mimo editor. Uživatel nahraje soubor a systém vloží správný odkaz do obsahu editoru, takže ihned po nahrání se zobrazí odkaz na soubor v obsahu editoru.

## Nový příspěvek

$$\sum_{i=1}^n 3n + 1$$

[Architecture.eps](#)

*Kurzíva* **Tučně**

body p strong

Soubor Choose File No file chosen Vložit soubor

☐ Anonymní příspěvek

Odkaz na uživatele:

Vytvořit

Obrázek 5.1: CKEditor na stránce pro vytvoření příspěvku

Pro interaktivní tvoření vzorců v jazyce  $\text{\LaTeX}$  bylo použito rozšíření „Equation Editor“ pro CKEditor. Toto řešení přináší omezení dané tvůrci, a to je maximálně 3000 vygenerovaných vzorců za den. Interaktivní editor po dokončení vzorce uloží do obsahu CKEditoru odkaz na server tvůrců, kde parametr v odkazu je zdrojový kód vzorce v jazyce  $\text{\LaTeX}$ . Žádný obrázek není uložen a pro každé zobrazení je nutné zaslat požadavek na server. Tímto způsobem by limit 3000 vzorců byl vyčerpán rychle. Toto rozšíření je vydáno pod licencí GPL (GNU General Public License), která říká:

„1. Můžete kopírovat a šířit doslovné kopie zdrojového kódu Programu tak, jak jste jej obdržel, a na libovolném médiu, za předpokladu, že na každé kopii viditelně a náležitě zveřejníte zmínku o autorských právech a absenci záruky; ponecháte nedotčené všechny zmínky vztahující se k této Licenci a k absenci záruky; a dáte každému příjemci spolu s Programem kopii této licence.

2.Můžete upravovat vaši kopii, či kopie Programu, anebo kterékoliv jeho části a tak vytvořit dílo založené na Programu, a kopírovat a šířit takové úpravy, či dílo, podle podmínek odstavce 1 výše, za předpokladu, že splníte všechny tyto podmínky:

- (a) Upravené soubory musíte opatřit zřetelnou zmínkou uvádějící, že jste soubory změnily s datem každé změny.
- (b) Musíte umožnit, aby jakékoliv vámi zveřejněné či šířené dílo, které jako celek nebo zčásti obsahuje Program nebo jakoukoli jeho část, popřípadě je z Programu nebo jeho části odvozeno, mohlo být jako celek bezplatně poskytnuto každé třetí osobě v souladu s ustanoveními této Licence.
- (c) Pokud upravený Program interaktivně čte povely, musíte zajistit, že při nejběžnějším způsobu jeho spuštění vytiskne nebo zobrazí hlášení zahrnující příslušnou zmínku o autorském právu a zmínku o tom, že neexistuje žádná záruka (nebo případně, že záruku poskytnete vy), a že uživatelé mohou za těchto podmínek Program šířit dále, a musí uživateli sdělit, jakým způsobem může nahlédnout do textu této Licence. (Výjimka: v případě, že sám Program je interaktivní, avšak žádné takové hlášení nevypisuje, nepožaduje se, aby vaše dílo založené na Programu takové hlášení vypisovalo.)”[17].

Protože lze rozšíření díky této licenci volně upravovat, je u vygenerovaného obrázku změněna adresa a místo serveru autorů *Equation Editor* vede odkaz na server Fituška2 na adresu, kde požadavky na vytvoření obrázku zpracovává servlet. Limit 3000 požadavků se čerpá pouze při vytváření vzorce v interaktivním editoru, ale už ne při jeho zobrazení v diskuzi. V budoucnu by bylo dobré toto rozšíření nahradit vlastním rozšířením a odstranit tak limit úplně.

Ukázka odkazu pro získání vygenerovaného obrázku pro vzorec vedoucí na server tvůrců

`latex.codecogs.com/gif.latex?%5Csum_%7Bi%3D1%7D%5E%7Bn%7D%203n&plus;1`



## Kapitola 6

# Implementace serverové části

Implementace proběhla podle vytvořeného návrhu až na odchylky, které se objevily u komponent využívajících služby třetí strany.

### 6.1 Služba Facebook

Původní návrh počítal s tím, že systém Fituška2 bude prohledávat skupiny na síti Facebook a bude z nich stahovat příspěvky, ukládat je a bude do skupin nahrávat příspěvky vytvořené v systému. Při návrhu tohoto systému byl vytvořen prototyp, na kterém byly otestovány požadavky, které má služba Facebook implementovat. Bohužel 4.4.2018 společnost Facebook změnila pravidla pro aplikace, které přistupují k síti Facebook (podrobnosti v dokumentu <https://developers.facebook.com/blog/post/2018/04/04/facebook-api-platform-product-changes/>). Domnívám se, že tato dramatická změna nastala kvůli novému nařízení Evropské Unie General Data Protection Regulation neboli GDPR<sup>1</sup>, které se snaží o lepší ochranu osobních dat. Od tohoto data musí každá aplikace, která chce přistupovat k příspěvkům ve skupinách, projít kontrolou pracovníky společnosti Facebook a musí být schválena. Samotná příprava aplikace na tuto kontrolu by vyžadovala spoustu práce a času, s čímž se při návrhu nepočítalo. Dále společnost Facebook pozastavila schvalování nových aplikací až do odvolání, pravděpodobně kvůli dokončení změn týkajících se GDPR. Toto pozastavení trvalo až do začátku měsíce května 2018. Z těchto důvodů není služba Facebook v novém systému plně implementována. Je implementován pouze základ služby, kdy lze službu zapnout a vypnout, služba může přijímat požadavky a zařadit je do fronty, vlákna mohou požadavky z fronty vybírat ale už je nemůžou zpracovat, a jsou definované některé typy požadavků v podobě tříd.

---

<sup>1</sup>„GDPR představuje nový právní rámec ochrany osobních údajů v evropském prostoru s cílem hájit co nejvíce práva občanů EU proti neoprávněnému zacházení s jejich daty včetně osobních údajů. GDPR se týká všech firem a institucí, ale i jednotlivců a online služeb, které zpracovávají data uživatelů. GDPR zavádí astronomické pokuty za porušování nových, přísnějších pravidel a nařizuje některým správcům nebo zpracovatelům osobních údajů zřídit nezávislou kontrolní funkci DPO (Data Protection Officer, tj. Pověřenec pro ochranu osobních údajů).“[13]

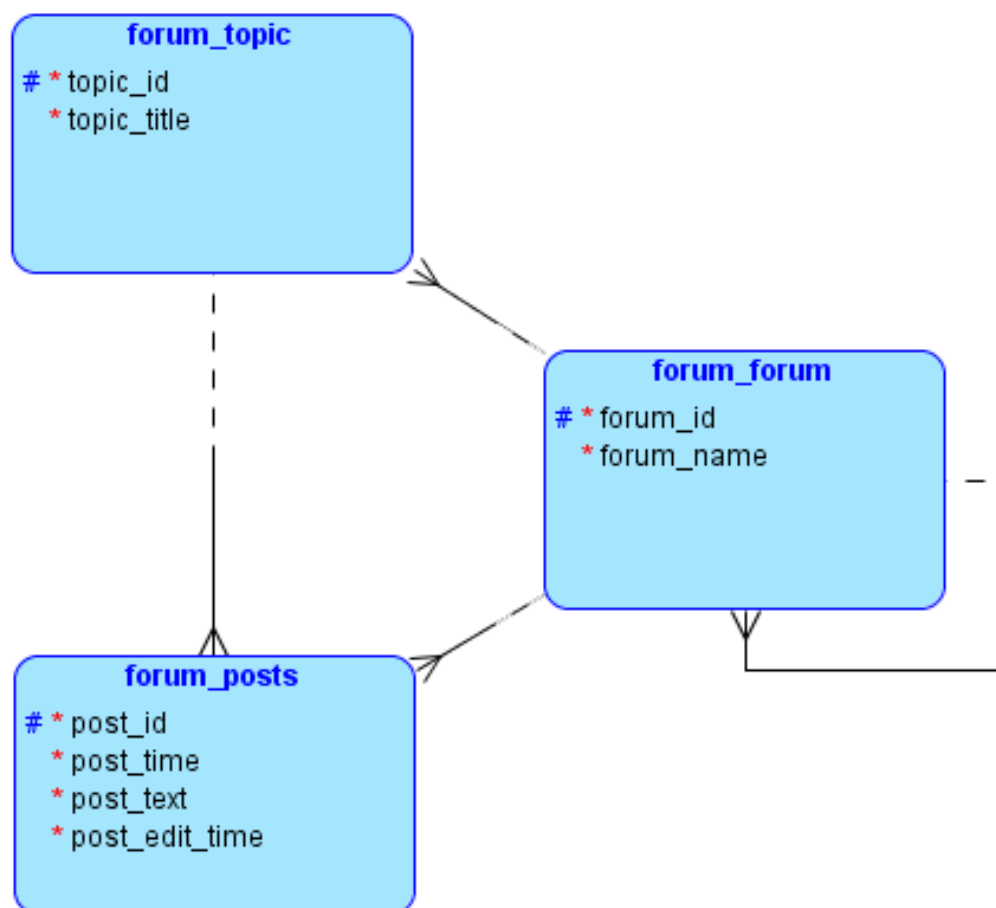
Protože integrace Fitušky2 se sociální sítí Facebook má být jedna z hlavních věcí, které mají přilákat studenty, aby nový systém používaly, bude nutné tuto službu implementovat hned, jak se situace se sítí Facebook ustálí a bude možné získat oprávnění k přístupu k příspěvkům ve skupinách.

## 6.2 Služba Google

Při návrhu služby Google jsem počítal s tím, že počet požadavků je omezen na den pro aplikace a pro uživatele v aplikaci (uživatelé se myslí celý systém Fituška2, ne koncoví uživatelé, kteří jsou do systému přihlášení). Tato omezení lze nastavit na jejich maximální hodnotu 1000000 požadavků na den. Při testování aplikace jsem ale narazil na omezení frekvence požadavků, které systém Google vyřídí za určitý čas (řádově asi vteřiny). V dokumentacích společnosti Google k daným službám jsem ale nenašel limity na toto omezení. Pokud tedy systém v navrženém stavu překročí toto omezení, spousta následujících požadavků nebude vyřízena. Naštěstí systém Google při chybové odpovědi specifikuje důvod chyby, takže lze rozpoznat, kdy chyba nastala z důvodů překročení frekvence požadavků. Proto bylo implementováno rozpoznávání chybové zprávy v odpovědi a funkce pro opětovné vložení požadavku do fronty s menší prioritou. Toto samotné řešení by způsobilo, že po prvním překročení limitu frekvence požadavků by systém Fituška2 donekonečna odesílal stejné požadavky, které by způsobovaly donekonečna chyby. Proto se konzumenti požadavků, po tom co je vrácena chyba překročení frekvence, na krátkou dobu zastaví a pokusí se tak snížit frekvenci požadavků. Doba, na kterou se konzument zastaví, musela být stanovena experimentálně, protože délka různých typů požadavků je různá a frekvence, kdy systém Google začne požadavky odmítat, není specifikována.

## 6.3 Migrace předchozí databáze

Během implementace byla provedena částečná migrace databáze současného systému Fituška. Protože tato databáze má rozdílné schéma od schématu databáze nového systému (náznak části schématu staré databáze je uveden na obrázku 6.1), bylo nutné transformovat data ze staré struktury na novou. Transformace se týkala pouze příspěvků a diskuzí, které se vážou k předmětům. Žádná jiná data nebyla ze staré databáze migrována. Na obrázcích 6.2 a 6.3 je uveden příklad, jak jsou strukturována data ve staré databázi a jak by data měla vypadat v nové databázi.



Obrázek 6.1: Náznak části struktury staré databáze.

```

├─ Předměty (2017/2018) (forum) .2 1BIT (forum)
│   └─ Zimní semestr (forum)
│       └─ IUS (forum)
│           └─ Kovar materialy (topic)
├─ Archiv
│   └─ Předměty (2007/2008) (forum)
│       └─ 1BIT (forum)
│           └─ Zimní semestr (forum)
│               └─ IUS (forum)
│                   └─ 4. projekt (topic)
  
```

Obrázek 6.2: Příklad dat uložených podle schématu staré databáze

```

IUS (course)
├─ Kovar materialy (discussion)
└─ 4. projekt (discussion)

```

Obrázek 6.3: Příklad 6.2 převeden podle schématu nové databáze

Tato transformace probíhala tak, že byla stará databáze zkopírována do databázového stroje, kde běžela i nová databáze. Na kopii staré databáze bylo možné provádět změny bez poškození celého současného systému Fituška.

Nejprve byly odstraněny záznamy z tabulky *forum\_forums*, které nejsou navázány na žádný další záznam v tabulce *forum\_forums*. Tyto záznamy tvořily strukturu, která v novém systému není potřeba.

```

DELETE F
FROM forum_forums F left join forum_topics T on (f.forum_id=t.forum_id)
WHERE T.topic_id is null

```

Další krok je odstranění duplicitních záznamů v tabulce *forum\_forums* a úprava cizích klíčů vedoucích z tabulky *forum\_topics* na odstraněné záznamy. Nejprve se vybere 1 záznam s daným jménem (v příkladu je to jméno „IAS“) a uloží se jeho primární klíč do proměnné *@id*. Následně se aktualizují všechny cizí klíče vedoucí na některý záznam v *forum\_forums* se jménem „IAS“ na uloženou hodnotu a smažou se všechny záznamy v *forum\_forums* se jménem „IAS“ s jiným než uloženým identifikátorem.

```

set @var='IAS';
set @id = (SELECT F.forum_id
FROM forum_forums F
WHERE F.forum_name=@var
ORDER BY F.forum_id LIMIT 1);
UPDATE forum_topics T
SET T.forum_id = @id
WHERE T.forum_id IN
(SELECT F.forum_id FROM forum_forums F where F.forum_name=@var);
DELETE FROM 'forum_forums' WHERE forum_name=@var AND forum_id<>@id

```

Předchozí sekvenci příkazů lze vygenerovat následujícím příkazem v jazyce BASH. Tento příkaz potřebuje jako vstup soubor „forumList.txt“, který obsahuje seznam všech unikátních jmen předmětů v tabulce *forum\_forums*. Vygenerované příkazy v jazyce SQL jsou uloženy do souboru „sqlcommands.txt“.

```

awk '{printf
"set @var=\'\''\'%s\'\'';\n
set @id = (SELECT F.forum_id FROM forum_forums F
WHERE F.forum_name=@var
ORDER BY F.forum_id LIMIT 1);\n

```

```

UPDATE forum_topics T SET T.forum_id = @id
WHERE T.forum_id IN
    (SELECT F.forum_id
     FROM forum_forums F
     WHERE F.forum_name=@var);\n
DELETE FROM 'forum_forums'
WHERE forum_name=@var AND forum_id<>@id;\n\n",
$0
}' ./forumList.txt > sqlcommands.txt

```

Pro vytvoření unikátních jmen diskuzí (záznamů v tabulce *forum\_topics*) byl před jméno diskuze přidán identifikátor záznamu. Pokud by toto nebylo provedeno, tak při vložení dat do nové databáze, by byl porušen unikátní klíč, který kontroluje unikátnost jmen diskuzí.

```

UPDATE forum_topics T
SET T.topic_title = concat(T.topic_id, T.topic_title)

```

Po těchto transformacích je již možné vložit data do nové databáze. Nejprve budou vloženy samotné předměty do tabulky *courses*.

```

INSERT INTO fituska.courses
SELECT F.forum_id," ",F.forum_name,null,null,null
FROM oldfituska.forum_forums F
WHERE (F.forum_name like '____' OR F.forum_name like '___') AND
      (F.forum_name NOT LIKE '1%' AND
       F.forum_name NOT LIKE '2%' AND
       F.forum_name NOT LIKE '3%' AND
       F.forum_name <> 'Brno'
       AND F.forum_name <> 'Akce')

```

Jako další jsou vloženy všechny diskuze.

```

INSERT INTO fituskatest.discussions
SELECT T.topic_id,T.topic_title,null,1,T.forum_id
FROM oldfituska.forum_topics T
JOIN fituskatest.courses C ON (T.forum_id=C.id)

```

A naposledy jsou vloženy samotné příspěvky.

```

INSERT INTO fituskatest.posts
SELECT P.post_id, P.post_text,
      FROM_UNIXTIME(P.post_time), FROM_UNIXTIME(P.post_time),
      1,0,0,'DISCUSSION',null,null,P.topic_id,null
FROM oldfituska.forum_posts P
WHERE P.topic_id IN
      (select D.id FROM fituskatest.discussions D)

```

# Kapitola 7

## Testování

Testování systému bylo rozděleno na 2 hlavní části. V první části jsem testoval aplikaci sám a jednalo se tedy o testování programátorem. V druhé části systém testovali koncoví uživatelé, tedy studenti.

### Log4j

Pro usnadnění testování jsou v systému použity logy pomocí technologie Log4j, která umožňuje provádět zápisy do logů s nastavenou úrovní<sup>1</sup>.

Výpisy komponent všech technologií, které Fituška2 používá, jsou přesměrovány ze standardního výstupu pomocí Log4j do logů. Protože tyto technologie na nižších úrovních (zejména na úrovni *DEBUG*) provádějí spousty zápisů, rozhodl jsem se vytvořit 2 logy. Do prvního logu se zapisuje jakýkoliv zápis s úrovní *WARN* a vyšší. V tomto logu lze tedy najít jakoukoliv chybu nebo varování, které systém zapsal. Druhý log slouží spíše pro ladění aplikace a jsou do něj přesměrovány všechny zápisy s úrovní *DEBUG* a vyšší ze tříd, které jsou v systému implementovány, přesněji řečeno všechny zápisy, které byly provedeny ze tříd umístěných v balíčku *cz.vut.fit.fituska*.

Ukázka logu, do kterého zapisoval nástroj Log4j

```
2018-05-14 10:48:45 ERROR SqlExceptionHandler:131 - Data truncated for
column 'schoolYear' at row 1
2018-05-14 10:48:17 WARN DocumentServiceImpl:65 - failed to initialize
calendar service
java.lang.IllegalThreadStateException
    at java.lang.Thread.start(Thread.java:708)
    at cz.vut.fit.fituska.services.googleServices.
documents.DocumentServiceImpl.
startThreads(DocumentServiceImpl.java:79)
// výpis celé výjimky
```

---

<sup>1</sup>Log4j používá 6 úrovní zápisů, které lze seřadit od nejzávažnější po nejméně závažné následovně: *FATAL*, *ERROR*, *WARN*, *INFO*, *DEBUG* a *TRACE*.

2018-05-21 17:22:01 INFO ASMemberSearcherService:36 - Modul

ASMemberSearcherService has been disabled

2018-05-21 17:23:15 DEBUG CalendarServiceImpl:89 - threads were ended

## Testování programátorem

V rámci testování programátorem byly zvoleny 3 druhy testů (testování komponent, integrační testování a systémové testování) a 2 testovací techniky (testování funkčnosti a destruktivní testování). Protože jsem tyto testy prováděl já se znalostmi celého softwaru a byly cílené na možné chyby ve vytvořené implementaci, jedná se o tzv. white-box testování.

Podle návrhu je systém jasně rozdělen na komponenty. Všechny tyto komponenty byly během implementace samostatně testovány k zjištění ověření funkčnosti. Komponenty, které patří do kategorie služeb, byly nejprve vytvořeny jako prototypy mimo systém, kde byly samostatně otestovány, a následně byly integrovány do systému. Po této integraci bylo provedeno integrační testování k ověření zachování funkčnosti komponent i celého systému.

Po dokončení implementace komponent bylo provedeno tzv. destruktivní testování komponent. Destruktivní testování se pokouší najít vstupy, které způsobí chybu v komponentě nebo v celém systému. Výsledky tohoto testování vedly k vytvoření dalších kontrol uživatelských vstupů. Nové kontroly pokrývaly nejen formát vstupů, ale i jejich délky, které by mohly způsobit problémy při vložení do databáze.

Poslední testování programátorem bylo systémové testování přes uživatelské rozhraní. V tomto testování jsem se pokusil simulovat typické chování uživatele v systému od jeho registrace přes vytváření příspěvků a dokumentů až po nastavení systému administrátorem.

## Testování uživateli

Po dokončení implementace prvních 3 iterací byla aplikace nasazena na školní server `ivs.fit.vutbr.cz`, kde byla k dispozici studentům k otestování a k připomínkování. Po nasazení jsem studenty požádal o otestování na jedné ze skupin na sociální síti Facebook. Podle dat v databázi se do aplikace úspěšně zaregistrovalo okolo 20-ti uživatelů, kteří si vyzkoušeli vytvoření diskuze, vložení příspěvku, nastavení profilu a prohlédli si strukturu webových stránek.

Bohužel se během tohoto testování ukázalo, že server `ivs.fit.vutbr.cz` není dostupný pro všechny studenty, a tak si nový systém nemohli vyzkoušet všichni zájemci. Výsledkem tohoto testu byly zprávy o 2 závažných chybách, ke kterým docházelo z důvodu chyby v uživatelském rozhraní, a nápady k vylepšení stránky zobrazující seznamy předmětů. Většinu připomínek jsem vyhověl a patřičně jsem tuto stránku upravil.

Dalším výstupem z uživatelského testování studenty byly logy, které zaznamenávaly chyby a varování, které se během testování objevily. Logy dohromady svou velikostí přesahovaly 1 MB a obsahovaly zápisy, které vedly k odhalení dalších chyb v serverové

části aplikace. Konkrétně byly odhaleny chyby v implementaci tříd, které komunikovaly s databází. Tyto chyby pravděpodobně vznikly, když jsem upravil schéma databáze, ale neupravil jsem třídy v serverové části aplikace.



# Kapitola 8

## Závěr

V této práci byly analyzovány hlavní studentské komunikační kanály, mezi které patří sociální síť Facebook, současný systém Fituška a Dokumenty Google. Během této analýzy jsem odhalil chyby, slabé stránky a silné stránky těchto kanálů. Tyto poznatky jsem využil při návrhu nového systému. Hlavním přínosem nového systému oproti současnému je přehlednější uživatelské rozhraní, integrace Kalendářů Google, sazba vzorců v jazyce L<sup>A</sup>T<sub>E</sub>X a lepší struktura dat.

Během návrhu systému vznikl seznam požadavků, návrh rolí v systému, detailní návrhy uživatelského rozhraní, popis architektury, schéma databáze a návrhy služeb založené na vytvořených prototypch. Při sběru požadavků byla využita analýza komunikačních kanálů a požadavky Studentské unie. Tento návrh společně s vybranými technologiemi bezpečně postačil pro vytvoření samotné implementace systému.

Během implementace došlo k nečekané události, kdy společnost Facebook změnila své podmínky pro vytváření aplikací přístupujících k její sociální síti, která znemožnila funkční implementaci integrace sociální sítě Facebook. V systému je implementován základ služby Facebook, který lze použít pro finální implementaci služby, až bude možné získat přístup k sociální síti.

Výběr jazyka Java a použitých technologií se ukázal jako správný. Podařilo se vytvořit systém, který lze snadno nasadit bez složité konfigurace a vybrané technologie pomohly usnadnit návrh i implementaci. Výsledná aplikace přesahuje velikost malé aplikace, takže se na ní vztahuje i obecná zvyklost použití jazyka Java a technologií Java EE pro větší aplikace.

Z implementačních iterací byly provedeny první 3 a byl vytvořen systém, který byl dán studentům k otestování. Na základě tohoto uživatelského testování vznikly návrhy ke změnám uživatelského rozhraní, které byly implementovány, a zprávy o chybách v uživatelském rozhraní a serverové části, které byly opraveny.

Výsledkem této práce je funkční systém, který lze ihned nasadit a začít používat. Stejně jako jakýkoliv software i tento systém bude nutné udržovat, opravovat a rozšiřovat, aby splňoval představy provozovatelů i uživatelů. Jako další rozšíření se nabízí požadavky ze 4. iterace, o které projeví uživatelé při reálném nasazení zájem. Další rozšíření by mohlo být získávání seznamu předmětů ze školních serverů bez nutnosti prohledávání

webových stránek školy, protože současné řešení je velmi citlivé na změny webových stránek školy. Pro toto rozšíření by bylo vhodné získat spolupráci školy, která by mohla poskytnout seznam předmětů ve vhodné formě pomocí některého ze svých serverů.

# Literatura

- [1] Alex, B.; Taylor, L.; Winch, R.; aj.: *Spring Security Reference*.  
<https://docs.spring.io/spring-security/site/docs/5.1.0.M1/reference/htmlsingle/>.
- [2] Ambler, S.: *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. 2002, ISBN 0-471-20282-7, citováno 11.01.2018.
- [3] Crume, J.; Mukhar, K.; Weaver, J. L.; aj.: *EJB fundamentals and session beans*.  
<https://www.javaworld.com/article/2071724/java-web-development/ejb-fundamentals-and-session-beans.html>, 2016, [Online; navštíveno 14.05.2018].
- [4] Dušek, D.: *Studentská skripta pro 3. ročník FIT*. <http://fit.fhfstudio.com/>, [Online; navštíveno 18.12.2017].
- [5] Evans, I.: *Your First Cup: An Introduction to the Java EE Platform*.  
<https://docs.oracle.com/javaee/7/firstcup/java-ee002.htm>, 2014.
- [6] Jendrock, E.; Cervera-Navarro, R.; Evans, I.; aj.: *The Java EE 6 Tutorial*.  
<https://docs.oracle.com/javaee/6/tutorial/doc/>, 2013, [Online; navštíveno 17.04.2018].
- [7] Joshi, R.: *Java design patterns: reusable solutions to common problems*. 2015, citováno 17.05.2018.
- [8] McQueeney, T.: *The Java EE Tutorial*.  
<https://javaee.github.io/tutorial/overview004.html>.
- [9] Peabody, B.: *Server-side I/O Performance: Node vs. PHP vs. Java vs. Go*.  
<https://www.toptal.com/back-end/server-side-io-performance-node-php-java-go>, [Online; navštíveno 13.05.2018].
- [10] Rychlý, M.: *Software Modelling Languages*. 2017, [Online; navštíveno 18.12.2017].
- [11] Sears, R.; van Ingen, C.; Gray, J.: *To BLOB or Not To BLOB: Large Object Storage in a Database or a Filesystem?* <https://www.microsoft.com/en-us/research/wp-content/uploads/2006/04/tr-2006-45.pdf>, 2006.

- [12] Teare, D.: *An Introduction To Ajax*. <http://www.oracle.com/technetwork/articles/grid/ajax-introduction-096831.html>, [Online; navštíveno 10.05.2018].
- [13] *Co je GDPR?* <https://www.gdpr.cz/gdpr/>, [Online; navštíveno 16.05.2018].
- [14] *Core J2EE Patterns - Data Access Object*. <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>.
- [15] *O webu*. <https://fitator.cz/about/>, [Online; navštíveno 17.12.2017].
- [16] *Prohlášení o právech a povinnostech*. <https://www.facebook.com/legal/terms/update>, [Online; navštíveno 17.12.2017].
- [17] *Obecná veřejná licence GNU v.2 (GNU GPL v.2)*. <http://www.gnugpl.cz/v2/>, 1991, [Online; navštíveno 11.05.2018].
- [18] Wikipedia contributors: *Apache Maven* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Apache\\_Maven&oldid=837443283](https://en.wikipedia.org/w/index.php?title=Apache_Maven&oldid=837443283), 2018, [Online; navštíveno 15. 05. 2018].
- [19] Wikipedie: *Cross-site request forgery* — *Wikipedie: Otevřená encyklopedie*. [https://cs.wikipedia.org/w/index.php?title=Cross-site\\_request\\_forgery&oldid=11615559](https://cs.wikipedia.org/w/index.php?title=Cross-site_request_forgery&oldid=11615559), 2014, [Online; navštíveno 16. 05. 2018].
- [20] Wikipedie: *PhpBB* — *Wikipedie: Otevřená encyklopedie*. <https://cs.wikipedia.org/w/index.php?title=PhpBB&oldid=14416710>, 2016, [Online; navštíveno 10.01.2018].
- [21] Wikipedie: *Producer–consumer problem* — *Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=Producer%E2%80%93consumer\\_problem&oldid=816492525](https://en.wikipedia.org/w/index.php?title=Producer%E2%80%93consumer_problem&oldid=816492525), 2017, [Online; navštíveno 13.01.2018].
- [22] Wikipedie: *Session* — *Wikipedie: Otevřená encyklopedie*. <https://cs.wikipedia.org/w/index.php?title=Session&oldid=14845052>, 2017, [Online; navštíveno 15. 05. 2018].
- [23] Wikipedie: *Framework* — *Wikipedie: Otevřená encyklopedie*. <https://cs.wikipedia.org/w/index.php?title=Framework&oldid=16073977>, 2018, [Online; navštíveno 15. 05. 2018].
- [24] Šebek, M.: *Implementace diskusní skupiny pro studenty*. bakalářská práce, FIT VUT v Brně, 2007.

## Příloha A

# Obsah přiloženého CD

	Aplikace	
	src .....	Zdrojové soubory aplikace
	pom.xml .....	Soubor definující závislosti pro nástroj Maven
	Práce .....	Obsahuje zdrojové soubory pro vytvoření PDF s textem práce
	xkrejc59-dip.pdf .....	Text diplomové práce
	dbinit.sql .....	Skript pro inicializaci databáze
	návod.txt .....	Návod na spuštění aplikace
	Fituska2.war .....	Sestavený systém (před nasazením si přečtěte návod)

# Příloha B

## Implementační iterace

### 1. iterace

- Uživatelské funkční požadavky
  - Uživatelské účty
    - \* přihlásit se
    - \* odhlásit se
    - \* vytvoření, editace a smazání účtu
    - \* základní nastavení (uživatelské jméno, jméno, příjmení, Facebook, Gmail, Skype, heslo, ročník a sebepopis)
    - \* přidělení a odebrání role
    - \* zasílání a přijímání zpráv
  - fórum
    - \* zobrazení předmětů, školních i mimoškolních diskuzí, písemek, diskuzí Studentské unie, kontaktů, detailu uživatele a zpráv
    - \* vytvoření, editace a smazání předmětové, školní a mimoškolní diskuze, zkoušky, diskuze Studentské unie
  - diskuze
    - \* vytvoření, editace, smazání a hodnocení (palce) příspěvků
    - \* vytvoření, editace, smazání zadání zkoušky
    - \* zobrazení příspěvků – autor (uživatelské jméno, ikona, role, rozšiřující role a sebe popis), zpráva, doba editace a počet palců
    - \* zobrazení řešení zkoušky – autor (uživatelské jméno, ikona, role, rozšiřující role a sebe popis), zprávy, doby editace a správnost (palce)
    - \* vytvoření, editace, smazání a hodnocení (palce) řešení zkoušky
    - \* vložení anonymního příspěvku
  - příspěvek
    - \* možný obsah

- text
  - obrázek
  - soubor
  - vzorec
  - odkaz na uživatele – metions
  - odkaz na webovou stránku
  - anketa
  - zdrojový kód
  - emotikony
- \* formát textu
  - barvy, velikost, tučné, podtržené, přeškrtnuté, kurzíva
- Systémové funkční požadavky
  - rutiny
    - \* import uživatelů ze školního serveru
    - \* získání seznamu předmětů ze školního serveru
    - \* získání odkazů na stránky předmětů školního serveru
  - ostatní
    - \* nastavení rutin
- Nefunkční požadavky
  - použití HTTPS a certifikátu
  - zamezení přístupu uživatelů k obsahu Fitušky 2.0 bez vytvořeného účtu
  - netransparentní zamezení přístupu uživatelů k obsahu bez patřičného oprávnění

## 2. iterace

- uživatelské
  - Uživatelské účty
    - \* nastavení sledovaných značek
    - \* nastavení sledovaných předmětů
    - \* nastavení sledovaných předmětů podle WISu
    - \* nastavení sledovaných školních diskuzí
    - \* nastavení sledovaných mimo školních diskuzí
    - \* nastavení sledovaných ostatních kalendářů
    - \* nastavení sledovaných kalendářů předmětů
  - fórum

- \* vytvoření, editace, smazání fórum značek
- \* zobrazení aktualit, kalendářů, nejbližších událostí,
- \* přiřazení a odebrání fórum a Facebook značky ke školní, předmětové a mimoškolní diskuzi
- diskuze
  - \* sledování a zrušení sledování školní, předmětové a mimoškolní diskuzi a zkoušky
  - \* kalendáře
  - \* vytvoření, smazání a editace kalendáře a událostí
- systémové
  - ostatní
    - \* nastavení práv k používání kalendáře
- Nefunkční požadavky
  - systém zamezí stavu kdy by nebyl aktivní žádný člen vedení studentské unie
  - systém nedovolí smazání uživatele s rolí Člen vedení Studentské unie pokud by jejich počet měl klesnout pod 2
  - pokud v systému zůstalo méně než 2 aktivní uživatelů s rolí Člen vedení Studentské unie, tak se povolí získat tuto roli jakémukoliv studentovi, který je člen studentské komory

### 3.iterace

- uživatelské
  - fórum
    - \* přiřazení a odebrání Facebook značky ke školní, předmětové a mimoškolní diskuzi
    - \* zaslání hromadné zprávy na Facebook (příspěvek do nastavených skupin)
  - Facebook
    - \* přidání, editace a odebrání Facebookových skupin
    - \* nastavení priority Facebookových skupin
- systémové
  - ostatní
    - \* odpojení modulů s cizím API
    - \* změna certifikátu bez nutnosti restartu serveru
  - rutiny



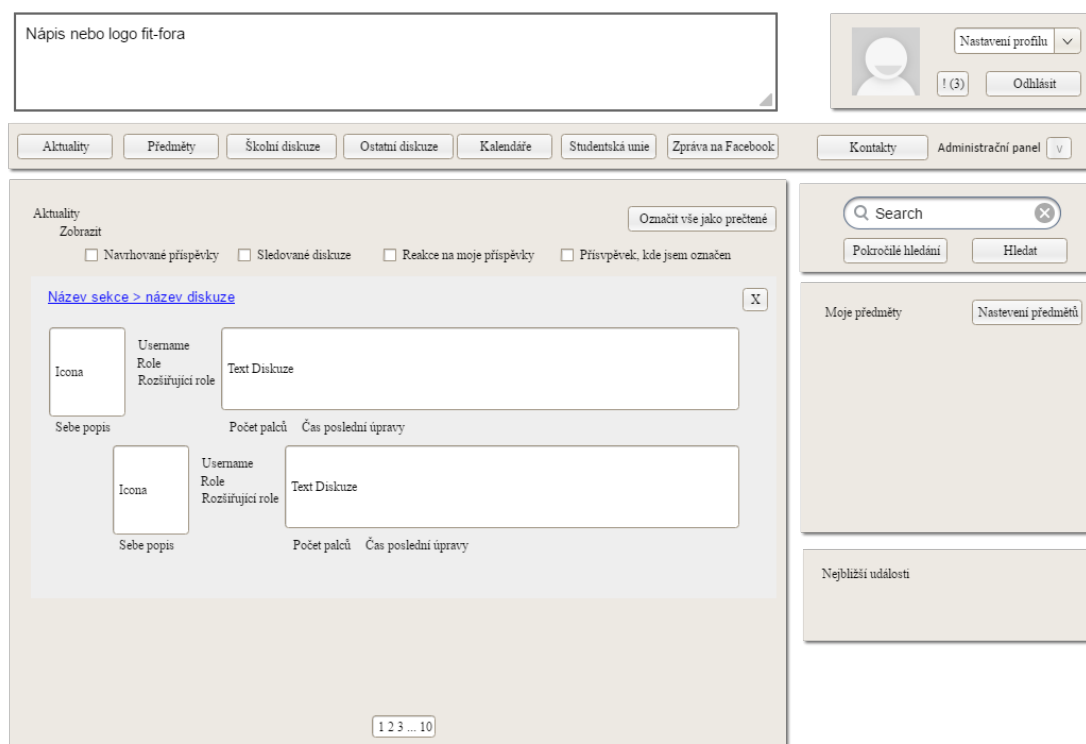
- \* smazání expirovaných uživatelů, událostí, příspěvků a Facebookových skupin
- Facebook
  - \* prohledávání Facebookových skupin a přenos příspěvků z Facebook na fórum
  - \* přenos příspěvků z fóra na Facebook pod účtem Fitušky 2.0

#### 4. iterace

- uživatelské
  - fórum
    - \* zobrazení datového skladu studentské unie
    - \* označení souboru značkou Pouze pro vedení
    - \* vytvoření, editace a smazání souboru v datovém skladu Studentské unie
    - \* autokorekce textu – čeština a slovenština
- systémové
  - ostatní
    - \* import uživatelů na požádání
    - \* vytvoření REST API pro mobilní aplikaci
    - \* notifikace do prohlížeče

## Příloha C

# Návrhy uživatelského rozhraní



Obrázek C.1: Návrh stránky s aktualitami

Nápis nebo logo fit-fora

Nastavení profilu

!

(3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Administrace fóra

Administrace facebooku

Upozornění:

Uložit

Tagy které mě zajímají

Nová zpráva

Nová žádost

Příspěvek ve sledovaném vlákně

Označení v příspěvku

Doporučený obsah

<

>

Upozornění zasílané na Facebook

Limit upozornění pro facebook

Input

Tagy které mě zajímají

Nová zpráva

Nová žádost

Příspěvek ve sledovaném vlákně

Označení v příspěvku

Doporučený obsah

<

>

Upozornění zasílané na email

Limit upozornění pro email

Input

Nastavení pro aktuality

Tagy které mě zajímají

1BIT

2BIT

<

>

Tagy které mě nezajímají

1MIT

2MIT

3BIT

DVI

Importovat Moje předměty z WISu

Moje předměty, které chci sledovat

Filtr

▼ Bakalářské

— IZP - Základní programování

— IDA - Diskrétní matematika

...

▼ Magisterské

— TIN - Teoretická informatika

— MAT - Matematické struktury v informatice

....

▼ Doktorské

— APD - Vybraná témata z analýzy a překladů jazyků

...

Moje předměty, které nechci sledovat

Filtr

▼ Bakalářské

— IOS - Operační systémy

— IFJ - Formální jazyky a překladače

...

▼ Magisterské

— FLP - Funkcionální a logické programování

— POS - Pokročilé operační systémy

....

▼ Doktorské

— JAD - Zkouška z jazyka anglického pro Ph.D.

...

Školní diskuze, které chci sledovat

Filtr

Školní diskuze, které nechci sledovat

Filtr

Koleje

Zápis předmětů

...

Ostatní diskuze, které chci sledovat

Filtr

Práce

Ostatní diskuze, které nechci sledovat

Filtr

Sport

Teambuilding

...

Moje předměty

Nastavení předmětů

Nejbližší události

79

Obrázek C.2: Návrh stránky s nastavením sledování

Nápis nebo logo fit-fora

Nastavení profilu

! (3) Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Předmět

Sledovat Zrušit sledování

Nový příspěvek

Diskuze

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del

Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců

Čas poslední úpravy

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del

Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců

Čas poslední úpravy

FB Diskuze

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del

Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců

Čas poslední úpravy

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del

Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců

Čas poslední úpravy

1 2 3 4 ... 10

Q Search

Pokročile hledání

Hledat

Moje předměty

Nastavení předmětů


Nejbližší události

Poznámka. Facebook diskuze bude barevně odlišena.

Obrázek C.3: Návrh stránky s diskuzí

80

Nápis nebo logo fit-fora



Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Administrace fóra

Administrace facebooku

Editace asociace diskuze s facebook tagy

Název diskuze

Název tagu

Název facebook skupiny

Název tagu

Název facebook skupiny

Název tagu

Název facebook skupiny

Název tagu

Název facebook skupiny

Přidat tag

Editace Přidání tagu

Název tagu

Přidané skupiny facebook k tagu

VUT FIT MIT 2016-2018 Teambuilding

VUT FIT MIT 2016-2018

Dostupné facebook skupiny

VUT FIT BIT 2015-2018

VUT FIT BIT 2014-2017

VUT FIT BIT 2013-2016

<<<

<

>

>>>


Uložit/Přidat

Poznámka. V dostupných skupinách barevně odlišit, které skupiny lze přidat a které ne(daný tag se v nich již vyhledává)

Obrázek C.4: Návrh stránky s editací asociace skupin sítě Facebook a diskuze

81

Nápis nebo logo fit-fora



Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Název nové diskuze

Text box

Tagy diskuze

Přidané tagy

1BIT

2BIT

Dostupné tagy

1MIT

2MIT

3BIT

DVI

Asociované facebook tagy:

Název tagu

Název tagu

Tag je vyhledáván ve facebook skupinách

Název facebook skupiny

Název facebook skupiny

Název facebook skupiny

Název facebook skupiny

Uložit

Poznamka. Po kliknutí na facebook tag se zobrazí skupiny, ve kterých se vyhledává.

Search

Pokročilé hledání

Hledat

Moje předměty


Nastavení předmětů

Nejbližší události

Obrázek C.5: Návrh stránky editace diskuze

82

Nápis nebo logo fit-fora



Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Název nové diskuze

Text box

Typ zkoušky

Půlsemestrálka

+

Text box

Vytvořit

Uložit

Poznámka. Komponenty v obdélníku se objeví až po kliknutí na tlačítko přidat ("+").

Poznámka. Po kliknutí na facebook tag se zobrazí skupiny, ve kterých se vyhledává.

Search

Pokročile hledání

Hledat

Moje předměty

Nastavení předmětů

Nejbližší události

Obrázek C.6: Návrh stránky editace zkoušky

Nápis nebo logo fit-fora

Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrální panel

Administrace fora

Administrace facebooku

Nastavení pro kalendář

Přidat mé předměty

Ostatní kalendáře

Koleje

Bakalářská práce

Diplomová práce

Kalendáře předmětů, které chci

Filtr

Bakalářské

IZP - Základní programování

IDA - Diskrétní matematika

...

Magisterské

TIN - Teoretická informatika

MAT - Matematické struktury v informatice

...

Doktorské

APD - Vybraná témata z analýzy a překladů jazyků

...

Kalendáře předmětů, které nechci

Filtr

Bakalářské

IOS - Operační systémy

IFJ - Formální jazyky a překladače

...

Magisterské

FLP - Funkcionální a logické programování

POS - Pokročilé operační systémy

...

Doktorské

JAD - Zkouška z jazyka anglického pro Ph.D.

...

Uložit

Search

Pokročilé hledání

Hledat

Moje předměty

Nastavení předmětů


Nejbližší události

Obrázek C.7: Návrh stránky s nastavením sledování kalendářů

84



Nápis nebo logo fit-fora



Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrální panel

Administrace fóra

Administrace facebooku

Název kalendáře:

☐ Kalendáře předmětů
 ☐ Kalendáře ostatní

Hledat

Název nového kalendáře:

Přidat

☐ Bakalářské  
☐ Magisterské  
☐ Doktorské

[Předmět - MIT - FLP - Funcionální a logické programování](#)

Přejmenovat

Smazat

[Ostatní - Koleje](#)

Přejmenovat

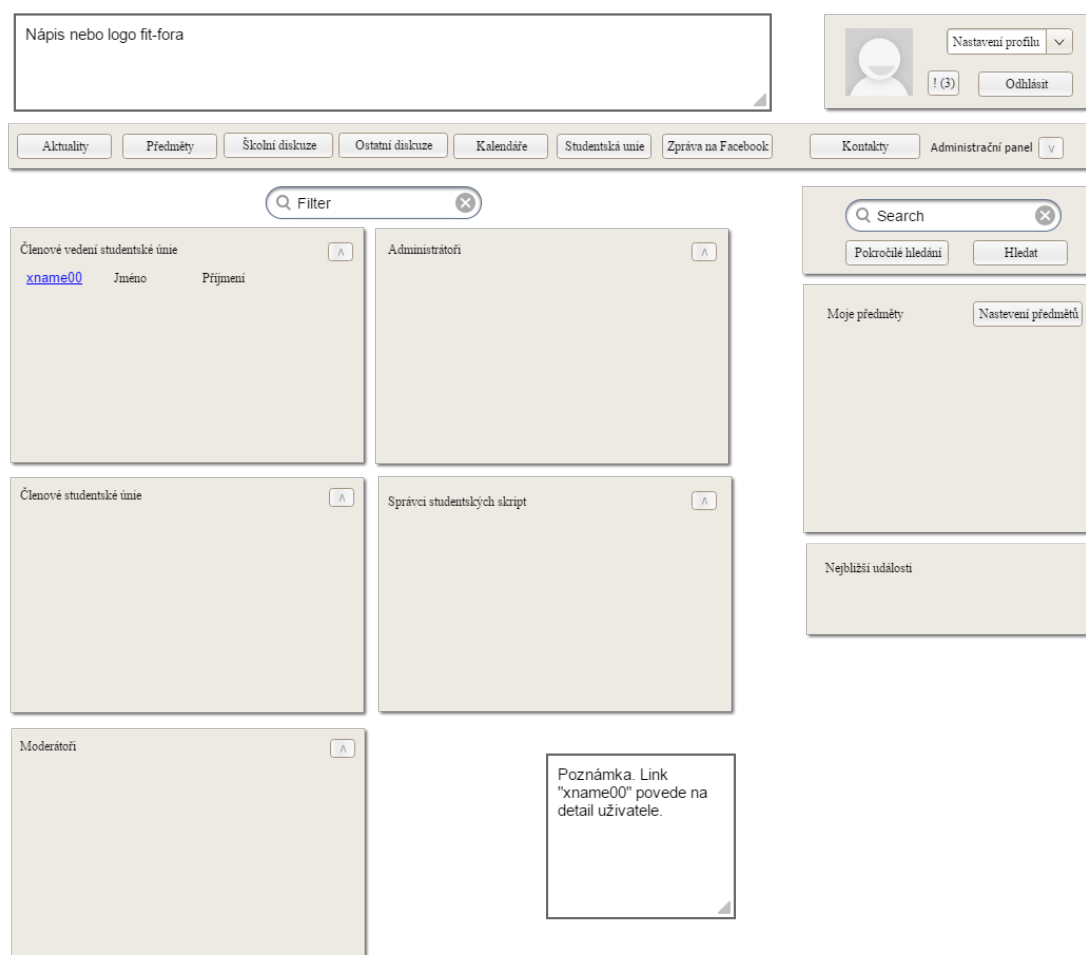
Smazat

1 2 3 ... 10

Poznámka. Odkazy povedou přímo na stránky google kalendáře mimo web fitfora


Obrázek C.8: Návrh stránky kalendářů





Obrázek C.10: Návrh stránky seznamu uživatelů

Nápis nebo logo fit-fora



Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendář

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Administrace fóra

Administrace facebooku

Import uživatelů

Datum provedení:

Text box

-

+

Provést teď

Text box

-

Importovat předměty

Datum provedení:

Text box

Provést teď

Smazání expirovaných uživatelů

Datum provedení:

Text box

Čas expirace po ukončení studia

Text box

Provést teď

Smazání expirovaných příspěvků

Datum provedení:

Text box

Čas expirace od vytvoření

Text box

Provést teď

Smazání expirovaných událostí

Datum provedení:

Text box

Čas expirace od vytvoření

Text box

Provést teď

Smazání expirovaných facebook skupin

Datum provedení:

Text box

Čas expirace od vytvoření

Text box


Provést teď

Uložit změny

Poznámka. Doporučená expirace příspěvků je 10 let.  
Doporučená expirace událostí je 2 roky.

Obrázek C.11: Návrh stránky s nastavením rutin

Nápis nebo logo fit-fora



Nastavení profilu ▾

! (3) Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel ^

Administrace fóra ▾

Administrace facebooku ▾

Přidat skupinu

VUT FIT MIT 2016-2018 Teambuilding

<https://www.facebook.com/groups/1777932395776133/>

neprioritní

Edit. Smazat

VUT FIT MIT 2016-2018

<https://www.facebook.com/groups/1064936046887651/>

prioritní

Edit. Smazat

Frekvence prohledávání prioritních skupin: 1 hodina ▾

Frekvence prohledávání neprioritních skupin: 1 hodina ▾

Přidání/editace skupiny

Název skupiny

Název:

URL:

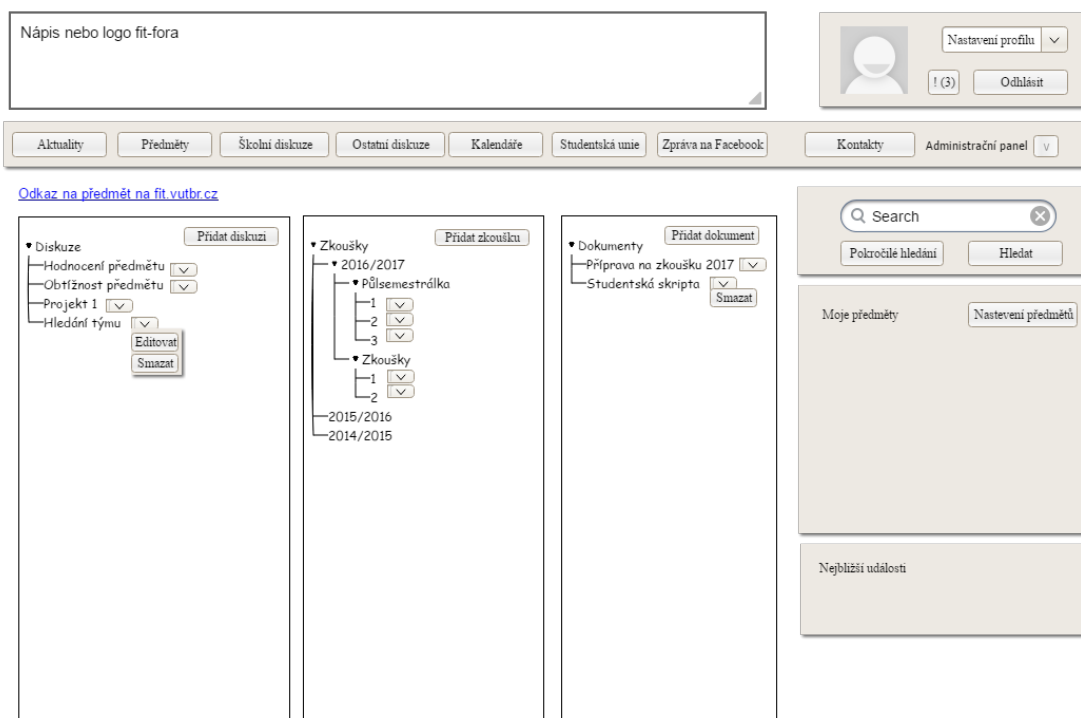
Priorita:  ▾

Přidat/Uložit

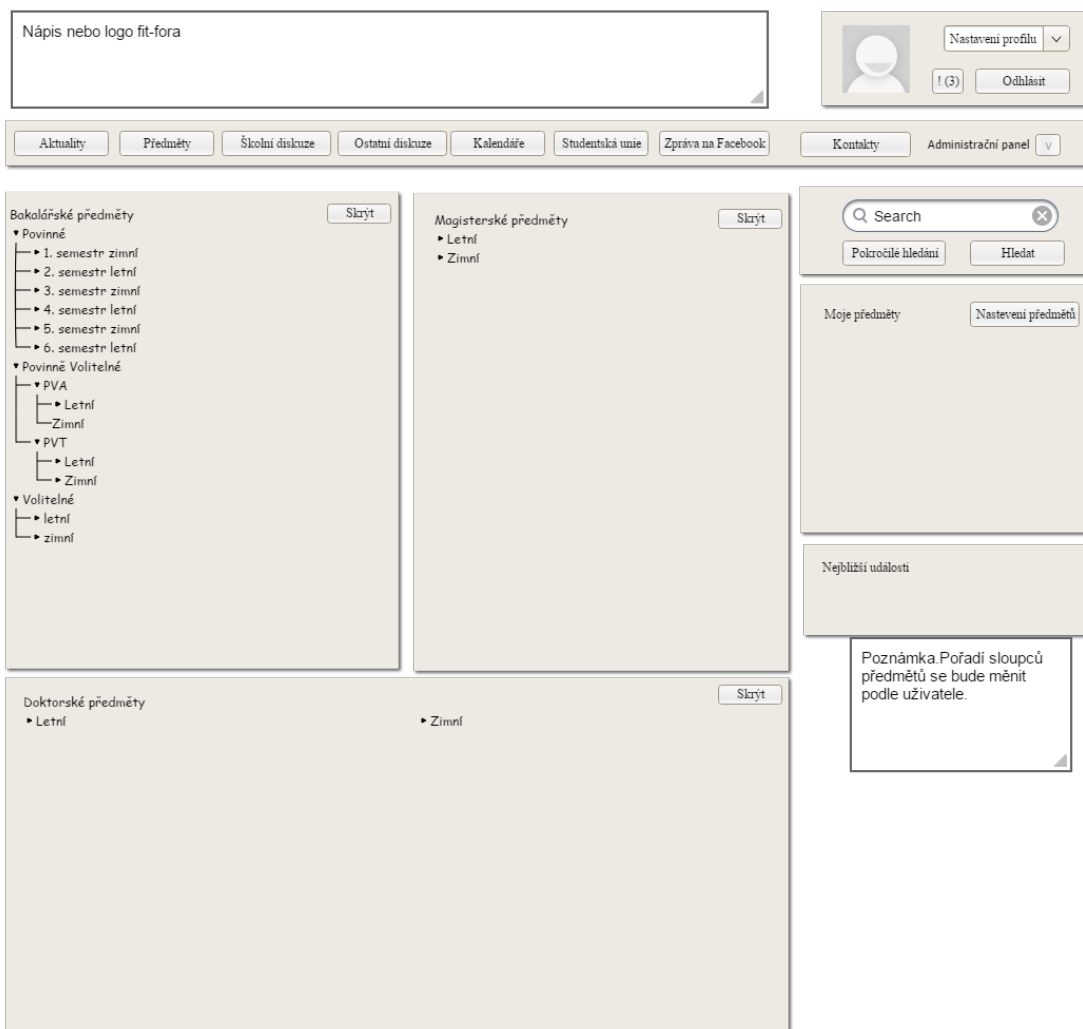
Poznámka. Po stisknutí edit se zobrazí detail na pravé straně.

Obrázek C.12: Návrh stránky s nastavením skupin sítě Facebook

89




Obrázek C.13: Návrh stránky předmětů



Obrázek C.14: Návrh stránky předmětu

Nápis nebo logo fit-fora



Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Administrace fóra

Administrace facebooku

1BIT

2BIT

3BIT

Přidat tag

Editovat

Smazat

Editovat

Smazat

Editovat

Smazat

Editace/Přidat tag

Název tag:

Text box


Uložit/Přidat

Poznámka. Podle toho zda uživatel kliknul na "Přidat tag" nebo na "Editovat" se zobrazí "Přidat tag" nebo "Editovat tag".

Obrázek C.15: Návrh stránky správa fórum značek



Nápis nebo logo fit-fora



Nastavení profilu

!

(3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendář

Studentská unie

Zpráva na Facebook

Kontakty

Administrální panel

Vhledávání

[Název sekce > název diskuze](#)

Icona

Username

Role

Rozlišující role

Text Diskuze

Sebe popis

Počet palců

Čas poslední úpravy

Icona

Username

Role

Rozlišující role

Text Diskuze

Sebe popis

Počet palců

Čas poslední úpravy

[Název sekce > název diskuze](#)

Icona

Username

Role

Rozlišující role

Text Diskuze

Sebe popis

Počet palců

Čas poslední úpravy

Icona

Username

Role

Rozlišující role

Text Diskuze

Sebe popis

Počet palců

Čas poslední úpravy

1 2 3 ... 10

Search

×

Hledat v:

☐

 Předměty
 

☐

 V názvech témat

☐

 Zkoušky
 

☐

 V textu témat

☐

 Studenská skripta
 

☐

 Ostatní diskuze

☐

 Škola
 

☐

 Ostatní

☐

 Studentská unie

Od:

Do:

Skrýt rozšířené hledání

Hledat

Moje předměty

Nastavení předmětů

Nejbližší události

Obrázek C.16: Návrh stránky vyhledávání

Nápis nebo logo fit-fora

Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrální panel

Název nové diskuze

Text box

Vytvořit

Forum tagy diskuze

Přidané tagy

1BIT

2BIT

Dostupné tagy

1MIT

2MIT

3BIT

DVI

Vybrané asociace diskuze s facebook tagy

Název tagu	Název facebook skupiny	Název tagu	Název facebook skupiny
	Název facebook skupiny		Název facebook skupiny
Název tagu	Název facebook skupiny	Název tagu	Název facebook skupiny
	Název facebook skupiny		Název facebook skupiny

Editace/Přidání tagu

Název tagu

Text box

Přidané skupiny facebook k tagu

VUT FIT MIT 2016-2018 Teambuilding

VUT FIT MIT 2016-2018

Dostupné facebook skupiny

VUT FIT BIT 2015-2018

VUT FIT BIT 2014-2017

VUT FIT BIT 2013-2016

Přidat tag

Moje předměty

Nastavení předmětů

Nejbližší události

Obrázek C.17: Návrh stránky vytvoření diskuze

Nápis nebo logo fit-fora

Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Název nové zkoušky

Text box

Typ zkoušky

Půlsemestrálka

+

Text box

Vytvořit

Vytvořit

Poznámka.

Komponenty v obdélníku se objeví až po kliknutí na tlačítko přidat ("+").

Q Search

Pokročile hledání

Hledat


Moje předměty

Nastavení předmětů

Nejbližší události

Obrázek C.18: Návrh stránky vytvoření zkoušky

Nápis nebo logo fit-fora



Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendář

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Administrace fóra

Administrace facebooku

Uživatelské jméno:

Jméno:

Příjmení:

Školní email:

Facebook:

Zobrazit

Gmail:

Zobrazit

Skype:

Zobrazit

Nové heslo:

Nové heslo znovu:

Ročník:

Role:


Další pravomoce:

Aministrátor, Člen stud. unie, ...

Sebepopis

Proděkan pro vzdělávací činnost v bakalářském studiu

Uložit



Změnit

Sada smajlíků:

Náhled smajlíků

228 x 185

Search

Pokročile hledání

Hledat

Moje předměty

Nastavení předmětů

Nejbližší události

Poznámka. Roli (student učitel) a další pravomoce nebude možné upravit bez patřičných práv.

Obrázek C.19: Návrh stránky se základním nastavení uživatelského účtu

Nápis nebo logo fit-fora

Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Předmět

Zobrazit diskuzi

Zobrazit řešení

Sledovat/Zrušit sledování

Zkouška

Del Edit

Skrýt

PDF/IMG/TEXT

Q Search

Pokročile hledání

Hledat

Moje předměty

Nastavení předmětů

Nejbližší události

Diskuze

Nové řešení

Nový příspěvek

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců

Čas poslední úpravy

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců

Čas poslední úpravy

Řešení

Icona

Username

Role

Rozšiřující role

Řešení

Del Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Správnost řešení

Čas poslední úpravy

Icona

Username

Role

Rozšiřující role

Text Diskuze

Del Edit

Sebe popis

Odpověď

Palec Up

Palec Down

Počet palců


Čas poslední úpravy

1 2 3 ... 10

Obrázek C.20: Návrh stránky zkoušky

Obrázek C.21: Návrh stránky s přehledem asociací mezi diskuzemi a skupinami sítě Facebook

Nápis nebo logo fit-fora



Nastavení profilu

! (3)

Odhlásit

Aktuality

Předměty

Školní diskuze

Ostatní diskuze

Kalendáře

Studentská unie

Zpráva na Facebook

Kontakty

Administrační panel

Zpráva

Publikovat ve facebook skupinách

VUT FIT MIT 2016-2018 Teambuilding

VUT FIT MIT 2016-2018

<<

<

>

>>

Dostupné facebook skupiny

VUT FIT BIT 2015-2018

VUT FIT BIT 2014-2017

VUT FIT BIT 2013-2016

Odeslat

Search

Pokročile hledání

Hledat

Moje předměty

Nastavení předmětů

Nejbližší události

Obrázek C.22: Návrh stránky pro odeslání hromadné zprávy sítí Facebook